Guide du développeur

AWS Key Management Service

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Key Management Service: Guide du développeur

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

AWS Key Management Service 1
Concepts 4
AWS KMS keys 5
Clés de client et clés AWS 6
Clés KMS de chiffrement symétrique 9
Clés KMS asymétriques 10
Clés KMS HMAC 11
Clés de données 11
Paires de clés de données 15
Alias 20
Magasins de clés personnalisés 21
Opérations cryptographiques 21
Identifiants clés () KeyId 23
Éléments de clé 26
Origine des éléments de clé 27
Spécifications de la clé 28
Utilisation de la clé 29
Chiffrement d'enveloppe 29
Contexte de chiffrement 31
Politique de clé 35
Octroi 35
Vérification de l'utilisation de clé KMS 35
Infrastructure de gestion des clés 36
Gestion de clés 37
Création de clés 37
Autorisations de création de clés KMS 40
Création de clés KMS de chiffrement symétriques 41
Utilisation des alias 46
À propos des alias 48
Gestion des alias 51
Utilisation d'alias dans vos applications 61
Contrôle de l'accès aux alias 63
Utilisation d'alias pour contrôler l'accès aux clés KMS 69
Recherche d'alias dans les journaux AWS CloudTrail 73

Affichage des clés ... 74
Affichage de clés KMS dans la console ... 75
Affichage des clés KMS avec l'API ... 90
Affichage de la configuration de chiffrement ... 98
Recherche de l'ID et de l'ARN d'une clé ... 99
Recherche du nom d'alias et de l'ARN d'alias ... 101
Modification des clés ... 103
Clés de balisage ... 105
À propos des balises dans AWS KMS ... 105
Gestion des balises de clé KMS dans la console ... 107
Gestion des balises de clés KMS avec les opérations API ... 108
Contrôle de l'accès aux balises ... 111
Utilisation de balises pour contrôler l'accès aux clés KMS ... 115
Activation et désactivation des clés ... 119
Activation et désactivation des clés KMS (console) ... 120
Activation et désactivation de clés KMS (API AWS KMS) ... 121
Rotating keys ... 122
Pourquoi faire pivoter les clés KMS ? ... 124
Comment fonctionne la rotation automatique des clés ... 125
Activation et désactivation de la rotation automatique des clés ... 128
Rotation manuelle des clés ... 130
Surveillance des clés ... 132
Outils de surveillance ... 133
Journalisation avec AWS CloudTrail ... 135
Surveillance avec CloudWatch ... 217
Surveillance avec Amazon EventBridge ... 230
Utilisation de CloudFormation modèles ... 232
Ressources AWS KMS dans les modèles AWS CloudFormation ... 233
En savoir plus sur AWS CloudFormation ... 234
Suppression des clés ... 235
À propos de la période d'attente ... 236
Suppression des clés KMS asymétriques ... 237
Suppression de clés multi-région ... 238
Suppression des clés KMS avec des éléments de clé importés ... 238
Contrôle de l'accès à une suppression de clé ... 238
Planification et annulation d'une suppression de clé ... 241

Création d'une alarme ... 245
Déterminer l'utilisation passée d'une clé KMS ... 248
Référence des états des clés ... 251
États de clé et types de clés KMS ... 252
Tableau d'état de clé ... 253
Authentification et contrôle d'accès ... 261
Concepts ... 262
Authentification ... 263
Autorisation ... 263
Authentification par des identités ... 263
Gestion des accès à l’aide de politiques ... 267
Ressources AWS KMS ... 270
Politiques de clé ... 271
Création d'une politique de clé ... 272
politique de clé par défaut ... 279
Affichage d'une politique de clé ... 294
Modification d'une politique de clé ... 297
Autorisations pour les services AWS ... 301
Politiques IAM ... 305
Présentation des politiques IAM ... 306
Bonnes pratiques pour les politiques IAM ... 307
Spécification de clés KMS dans les instructions de politique IAM ... 310
Autorisations requises pour utiliser la console AWS KMS ... 313
Politique gérée AWS pour les utilisateurs expérimentés ... 313
Exemples ... 315
Octrois ... 321
À propos des octrois ... 322
Concepts d'octroi ... 323
Bonnes pratiques ... 328
Création d'octrois ... 329
Gestion des octrois ... 338
Point de terminaison d’un VPC ... 343
Considérations relatives aux points de terminaison de VPC AWS KMS ... 344
Création d'un point de terminaison de VPC pour AWS KMS ... 344
Connexion à un point de terminaison VPC ... 345
Contrôle de l'accès à votre point de terminaison d'un VPC ... 346

Utilisation d'un point de terminaison VPC dans une déclaration de politique 350
Journalisation de votre point de terminaison d'un VPC 353
Clés de condition 355
AWS clés de condition globales 355
AWS KMS clés de condition 358
AWS KMS clés de condition pour AWS Nitro Enclaves 425
Utilisation du contrôle d'accès basé sur les attributs (ABAC) 429
Clés de condition ABAC pour AWS KMS 430
Des balises ou des alias ? 433
Résolution des problèmes liés à l'ABAC pour AWS KMS 435
Accès intercomptes 439
Étape 1 : ajouter une déclaration de politique de clé dans le compte local 441
Étape 2 : ajouter des politiques IAM dans le compte externe 445
Création de clés KMS que d'autres comptes peuvent utiliser 446
Autoriser l'utilisation de clés KMS externes avec Services AWS 448
Utilisation de clés KMS dans d'autres comptes 449
Rôles liés à un service 449
Autorisations des rôles liés à un service pour les magasins de clés personnalisés AWS KMS 450
Autorisations des rôles liés à un service pour les clés multi-région AWS KMS 451
Mises à jour AWS KMS vers des politiques gérées par AWS 451
TLS post-quantique hybride 452
À propos de Post-Quantum TLS 454
Comment l'utiliser 455
Configuration 456
Comment le tester 457
En savoir plus 458
Détermination de l'accès 458
Examen de la politique de clé 459
Examen des politiques IAM 462
Examen des octrois 464
Résolution des problèmes de clé d'accès 465
Référence des autorisations 473
Descriptions des colonnes 518
Test des autorisations 520
Qu'est-ce que c'est DryRun ? 521

Spécification DryRun à l'aide de l'API 522
Clés à usage spécial 523
Choix d'un type de clé KMS 524
Sélection de l'utilisation des clés 526
Sélection des spécifications de la clé 528
Clés asymétriques 530
Clés KMS asymétriques 531
Création de clés KMS asymétriques 533
Téléchargement de clés publiques 538
Identification des clés KMS asymétriques 542
Spécifications de clés asymétriques 547
Clés HMAC 560
Spécifications de clé pour les clés KMS HMAC 563
Création de clés HMAC 564
Contrôle de l'accès aux clés HMAC 569
Affichage de clés HMAC 570
Clés multi-région 571
Considérations sur la sécurité pour les clés multi-région 574
Fonctionnement des clés multi-région 575
Concepts 579
Contrôle de l'accès 582
Création de clés multi-régions 590
Affichage des clés multi-régions 601
Gestion des clés multi-régions 606
Importation des éléments de clé dans des clés multi-régions 612
Suppression de clés multi-régions 616
Éléments de clé importés 629
Planification pour importer des éléments de clé 632
Gestion des éléments de clé importés 640
Étape 1 : création d'une clé KMS sans élément de clé 647
Étape 2 : Téléchargement de la clé publique d’encapsulage et du jeton d'importation 651
Étape 3 : Chiffrement des éléments de clé 659
Étape 4 : Importation des éléments de clé 669
Magasins de clés personnalisés 672
AWS CloudHSM magasins clés 675
Magasins de clés externes 745

Référence des types de clés ... 886
Tableau des types de clé ... 887
Tableau des fonctionnalités spéciales ... 893
Sécurité ... 903
Protection des données ... 904
Protection des éléments de clé ... 904
Chiffrement des données ... 905
Trafic inter-réseaux ... 907
Gestion des identités et des accès ... 908
Journalisation et surveillance ... 909
Validation de la conformité ... 910
Documents de conformité et de sécurité ... 910
En savoir plus ... 911
Résilience ... 912
Isolement régional ... 912
Conception à locataires multiples ... 913
Bonnes pratiques de résilience dans AWS KMS ... 913
Sécurité de l'infrastructure ... 914
Isolation des hôtes physiques ... 915
Bonnes pratiques de sécurité ... 916
Quotas ... 917
Quotas de ressources ... 917
AWS KMS keys :100 000 ... 918
Alias par clé KMS : 50 ... 918
Octrois par clé KMS : 50 000 ... 919
Taille du document de stratégie de clé : 32 Ko ... 919
Quota de ressources des magasins de clés personnalisés : 10 ... 920
Quotas de demande ... 920
Demander des quotas pour chaque opération d'API AWS KMS ... 921
Application des quotas de demande ... 928
Quotas partagés pour les opérations de chiffrement ... 929
Demandes d'API effectuées en votre nom ... 930
Demandes entre comptes ... 931
Quotas de demandes de magasin de clés personnalisé ... 931
Limitation des demandes ... 933
Comment les services AWS utilisent AWS KMS ... 935

AWS CloudTrail 936
Comprendre quand votre clé KMS est utilisée 937
Amazon DynamoDB 944
Amazon Elastic Block Store (Amazon EBS) 944
Chiffrement Amazon EBS 945
Utilisation des clés KMS et des clés de données 945
Contexte du chiffrement Amazon EBS 946
Détection des défaillances Amazon EBS 947
Utilisation de AWS CloudFormation pour créer des volumes Amazon EBS chiffrés 948
Amazon Elastic Transcoder 948
Chiffrement du fichier d'entrée 948
Déchiffrement du fichier d'entrée 949
Chiffrement du fichier de sortie 951
Protection du contenu HLS 953
Contexte de chiffrement Elastic Transcoder 954
Amazon EMR 954
Chiffrement des données dans le système de fichiers EMR (EMRFS) 955
Chiffrement des données sur les volumes de stockage de nœuds de Cluster 958
Contexte de chiffrement 959
AWS Nitro Enclaves 960
Comment appeler des API AWS KMS pour une enclave Nitro 962
Clés de condition AWS KMS pour AWS Nitro Enclaves 963
Demandes de surveillance pour les enclaves Nitro 967
Amazon Redshift 972
Chiffrement Amazon Redshift 972
Contexte de chiffrement 973
Amazon Relational Database Service (Amazon RDS) 973
AWS Secrets Manager 974
Amazon Simple Email Service (Amazon SES) 974
Présentation du chiffrement Amazon SES à l'aide d'AWS KMS 975
Contexte du chiffrement Amazon SES 976
Autoriser Amazon SES à utiliser votre AWS KMS key 977
Récupération et déchiffrement des messages électroniques 978
Amazon Simple Storage Service (Amazon S3) 979
AWS Systems Manager Parameter Store 979
Protection des paramètres standard de chaîne sécurisée 981

Protection avancée des paramètres de chaîne sécurisée 984
Définition d'autorisations de chiffrement et de déchiffrement des valeurs de paramètres 987
Contexte de chiffrement Parameter Store 989
Résolution des problèmes de clés KMS dans Parameter Store 992
Amazon WorkMail 992
WorkMail Présentation d'Amazon 993
WorkMail Chiffrement Amazon 993
Autoriser l'utilisation de la clé KMS 997
Contexte WorkMail de chiffrement Amazon 999
Surveillance de WorkMail l'interaction d'Amazon avec AWS KMS 1000
WorkSpaces 1003
Vue d'ensemble du WorkSpaces chiffrement à l'aide de AWS KMS 1003
WorkSpaces contexte de chiffrement 1004
WorkSpaces Autorisation d'utiliser une clé KMS en votre nom 1005
Programmation de l'API AWS KMS 1008
Création d'un client 1008
Utilisation de clés 1010
Création d'une clé KMS 1010
Génération d'une clé de données 1012
Affichage d'un AWS KMS key 1016
Obtention des ID de clé et des ARN 1019
Activer AWS KMS keys 1021
Désactivation de AWS KMS key 1024
Utilisation des alias 1027
Création d'un alias 1027
Établissement de la liste des alias 1030
Mise à jour d'un alias 1035
Suppression d'un alias 1039
Chiffrement et déchiffrement des clés de données 1041
Chiffrement d'une clé de données 1042
Déchiffrement d'une clé de données 1045
Rechiffrement d'une clé de données sous une autre AWS KMS key 1049
Utilisation de politiques de clé 1054
Établissement de la liste des politiques de clé 1054
Obtention d'une politique de clé 1057
Définition d'une politique de clé 1060

Utilisation d'octrois 1067
Création d'un octroi 1067
Affichage d'un octroi 1071
Abandon d'un octroi 1077
Révocation d'un octroi 1079
Tester vos appels d’API AWS KMS 1083
Qu'est-ce que c'est DryRun ? 521
Spécification DryRun à l'aide de l'API 522
cohérence à terme AWS KMS 1085
Références 1086
Historique du document 1088
Mises à jour récentes 1088
Mises à jour antérieures 1093
..... mxcviii

AWS Key Management Service

AWS Key Management Service (AWS KMS) est un service géré qui facilite la création et le contrôle des clés de chiffrement utilisées pour créer et contrôler vos données. AWS KMS utilise des modules de sécurité matérielle (HSM) pour protéger et valider vos AWS KMS keys en vertu du programme de validation du module de chiffrement FIPS 140-2. Les régions Chine (Beijing) et Chine (Ningxia) ne prennent pas en charge le programme de validation des modules cryptographiques FIPS 140-2. AWS KMS utilise des HSM certifiés OSCCA pour protéger les clés KMS dans les régions chinoises.

AWS KMS est intégré à la plupart des autres services AWS qui chiffrent vos données. AWS KMS est également intégré à AWS CloudTrail pour journaliser l'utilisation de vos clés KMS pour les besoins d'audit, de réglementation et de conformité.

Vous pouvez utiliser l'API AWS KMS pour créer et gérer des clés KMS et des fonctionnalités spéciales, telles que des magasins de clés personnalisés, et utiliser des clés KMS dans les opérations de chiffrement. Pour plus d'informations, consultez la référence d'API AWS Key Management Service.

Vous pouvez créer et gérer vos AWS KMS keys :

- Créer, modifier et afficher des clés KMS symétriques et asymétriques, y compris des clés HMAC.
- Contrôlez l'accès à vos clés KMS à l'aide de politiques de clé, de politiques IAM et d'octrois. AWS KMS prend en charge le contrôle d'accès basé sur les attributs (ABAC). Vous pouvez également affiner les politiques en utilisant des clés de condition.
- Créez, supprimez, répertoriez et mettez à jour des alias, noms conviviaux pour vos clés KMS. Vous pouvez également utiliser des alias pour contrôler l'accès à vos clés KMS.
- Balisez vos clés KMS pour l'identification, l'automatisation et le suivi des coûts. Vous pouvez également utiliser des balises pour contrôler l'accès à vos clés KMS.
- Activez et désactivez les clés KMS.
- Activez et désactivez la rotation automatique du contenu de chiffrement dans une clé KMS.
- Supprimez des clés KMS pour terminer le cycle de vie des clés.

Vous pouvez utiliser vos clés KMS dans des opérations de chiffrement. Pour obtenir des exemples, consultez Programmation de l'API AWS KMS.

- Chiffrez, déchiffrez et chiffrez à nouveau des données à l'aide de clés KMS symétriques ou asymétriques.
- Signez et vérifiez les messages avec des clés KMS asymétriques.
- Générez des clés de données symétriques exportables et des paires de clés de données asymétriques.
- Générez et vérifiez des codes HMAC.
- Générez des nombres aléatoires qui conviennent pour les applications de chiffrement.

Vous pouvez également utiliser les fonctionnalités avancées de AWS KMS.

- Créer des clés multi-régions, qui agissent comme des copies de la même clé KMS dans différentes Régions AWS.
- Importer les éléments cryptographiques dans une clé KMS.
- Créez des clés KMS dans un magasin de clés AWS CloudHSM soutenu par votre cluster AWS CloudHSM.
- Créez des clés KMS dans un magasin de clés externe soutenu par vos clés cryptographiques à l'extérieur d'AWS.
- Vous connecter directement à AWS KMS via un point de terminaison privé dans votre VPC.
- Utilisez le protocole TLS post-quantique hybride pour fournir un chiffrement prospectif en transit pour les données que vous envoyez à AWS KMS.

En utilisant AWS KMS, vous obtenez plus de contrôle sur l'accès aux données que vous chiffrez. Vous pouvez utiliser les fonctions de gestion de clés et de chiffrement directement dans vos applications ou via les services AWS intégrés à AWS KMS. Que vous écriviez des applications pour AWS ou que vous utilisiez des services AWS, AWS KMS vous permet de contrôler qui peut utiliser vos AWS KMS keys et accéder à vos données chiffrées.

AWS KMS s'intègre à AWS CloudTrail, un service qui fournit des fichiers de journaux dans votre compartiment Amazon S3 désigné. En utilisant, CloudTrail vous pouvez surveiller et étudier comment et quand vos clés KMS ont été utilisées et qui les a utilisées.

AWS KMS dans Régions AWS

Les Régions AWS dans lesquelles AWS KMS est pris en charge sont répertoriées dans AWS Key Management Service Endpoints and Quotas. Si une fonction AWS KMS n'est pas prise en charge

dans une Région AWS que AWS KMS prend en charge, la différence régionale est décrite dans la rubrique sur la fonction.

Tarification d'AWS KMS

Comme avec d'autres produits AWS, l'utilisation de AWS KMS ne nécessite pas de contrats ou d'achats minimum. Pour plus d'informations sur la tarification AWS KMS, consultez Tarification AWS Key Management Service.

Contrat de niveau de service

AWS Key Management Service est soutenu par un contrat de niveau de service (SLA) qui définit la politique de disponibilité de notre service.

En savoir plus

- Pour de plus amples informations sur les termes et concepts utilisés dans AWS KMS, veuillez consulter Concepts AWS KMS.
- Pour plus d'informations sur l'API AWS KMS, veuillez consulter la référence de l'API AWS Key Management Service. Pour obtenir des exemples dans différents langages de programmation, veuillez consulter Programmation de l'API AWS KMS.
- Pour apprendre à utiliser les modèles AWS CloudFormation pour créer et gérer des clés et des alias, consultez Création de ressources AWS KMS avec AWS CloudFormation et la Référence de type de ressource AWS Key Management Servicedans le Guide de l'utilisateur AWS CloudFormation.
- Pour obtenir des informations techniques détaillées sur la façon dont AWS KMS utilise le chiffrement et sécurise les clés KMS, veuillez consulter AWS Key Management Service Détails cryptographiques. La documentation des détails cryptographiques ne décrit pas comment AWS KMS travaille dans les régions Chine (Beijing) et Chine (Ningxia).
- Pour obtenir une liste de points de terminaison AWS KMS, y compris les points de terminaison FIPS, dans chaque Région AWS, veuillez consulter Points de terminaison de service dans la rubrique AWS Key Management Service de la Références générales AWS.
- Pour obtenir de l'aide sur les questions relatives à AWS KMS, veuillez consulter le forum de discussion AWS Key Management Service.

AWS KMS dans les kits AWS SDK

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

Concepts AWS KMS

Apprenez les termes et les concepts de base utilisés dans AWS Key Management Service (AWS KMS) et la manière dont ces concepts contribuent ensemble à protéger vos données.

Rubriques

- AWS KMS keys
- Clés de client et clés AWS
- Clés KMS de chiffrement symétrique
- Clés KMS asymétriques
- Clés KMS HMAC
- Clés de données
- Paires de clés de données
- Alias
- Magasins de clés personnalisés
- Opérations cryptographiques
- Identifiants clés () KeyId
- Éléments de clé
- Origine des éléments de clé
- Spécifications de la clé
- Utilisation de la clé
- Chiffrement d'enveloppe
- Contexte de chiffrement

- Politique de clé
- Octroi
- Vérification de l'utilisation de clé KMS
- Infrastructure de gestion des clés

AWS KMS keys

Les AWS KMS keys (clés KMS) sont la ressource principale dans AWS KMS. Vous pouvez utiliser une clé KMS pour chiffrer, déchiffrer et chiffrer à nouveau des données. Elle peut également générer des clés de données que vous pouvez utiliser en dehors de AWS KMS. En règle générale, vous utiliserez des clés KMS de chiffrement symétriques, mais vous pouvez créer et utiliser des clés KMS asymétriques pour le chiffrement ou la signature et créer et utiliser des clés KMS HMAC pour générer et vérifier des balises HMAC.

> **Note**
>
> AWS KMS remplace le terme clé principale client (CMK) par AWS KMS key et clé KMS. Le concept n'a pas changé. Pour éviter les changements de rupture, AWS KMS conserve quelques variations de ce terme.

Une AWS KMS key est une représentation logique d'une clé cryptographique. Une clé KMS contient des métadonnées, telles que l'ID de clé, Spécifications de la clé, Utilisation de la clé, date de création, description et l'État de clé. Plus important encore, il contient une référence aux éléments de clé qui sont utilisés lorsque vous exécutez des opérations cryptographiques avec la clé KMS.

Vous pouvez créer une clé KMS avec des éléments de clé cryptographiques générés dans des modules de sécurité matérielle validés FIPS AWS KMS. Les éléments de clé des clés symétriques KMS et les clés privées des clés asymétriques KMS ne laissent jamais AWS KMS non chiffré. Pour utiliser ou gérer vos clés KMS, vous devez utiliser AWS KMS. Pour de plus amples informations sur la création et la gestion des clés KMS, veuillez consulter Gestion de clés . Pour de plus amples informations sur l'utilisation des clés KMS, veuillez consulter la Référence d'API AWS Key Management Service.

Par défaut, AWS KMS crée les éléments de clé pour une clé KMS. Vous ne pouvez pas extraire, exporter, afficher ou gérer ces éléments de clé. La seule exception est la clé publique d'une paire de clés asymétriques, que vous pouvez exporter pour l'utiliser en dehors de AWS. De plus, vous ne

pouvez pas supprimer ces éléments de clé ; vous devez supprimer la clé KMS. Vous pouvez toutefois importer vos propres éléments de clé dans une clé KMS, ou utiliser un magasin de clés personnalisé pour créer des clés KMS qui utilisent des éléments de clé de votre cluster AWS CloudHSM, ou des éléments de clé dans un gestionnaire de clés externe que vous possédez et gérez en dehors d'AWS.

AWS KMS prend également en charge les clés multi-région, qui vous permettent de chiffrer les données dans une Région AWS et les déchiffrer dans une autre Région AWS.

Pour de plus amples informations sur la création et la gestion des clés KMS, veuillez consulter Gestion de clés . Pour de plus amples informations sur l'utilisation des clés KMS, veuillez consulter la Référence d'API AWS Key Management Service.

Clés de client et clés AWS

Les clés KMS que vous créez sont des clés gérées par le client. Les Services AWS qui utilisent des clés KMS pour chiffrer vos ressources de service créent généralement des clés automatiquement. Les clés KMS que les Services AWS créent dans votre compte AWS sont des Clés gérées par AWS. Les clés KMS que les Services AWS créent dans un compte de service sont des Clés détenues par AWS.

Type de clé KMS	Peut afficher les métadonnées de clés KMS	Peut gérer une clé KMS	Utilisée uniquement pour mon Compte AWS	Rotation automatique	Tarification
Clé gérée par le client	Oui	Oui	Oui	Facultatif. Chaque année (environ 365 jours)	Frais mensuels (au prorata horaire) Frais par utilisation
Clé gérée par AWS	Oui	Non	Oui	Obligatoire. Chaque année (environ 365 jours)	Aucun frais mensuel Frais par utilisation (certains

Type de clé KMS	Peut afficher les métadonnées de clés KMS	Peut gérer une clé KMS	Utilisée uniquement pour mon Compte AWS	Rotation automatique	Tarification
					Services AWS paient ces frais pour vous)
Clé détenue par AWS	Non	Non	Non	Varie	Pas de frais

Les services AWS qui s'intègrent à AWS KMS diffèrent dans leur prise en charge des clés KMS. Certains services AWS chiffrent vos données par défaut avec une Clé détenue par AWS ou une Clé gérée par AWS. Certains services AWS prennent en charge les clés gérées par le client. D'autres services AWS prennent en charge tous les types de clés KMS pour vous permettre de bénéficier de la simplicité d'utilisation d'une Clé détenue par AWS, de la visibilité d'une Clé gérée par AWS ou du contrôle d'une clé gérée par le client. Pour plus d'informations sur les options de chiffrement proposées par un service AWS, consultez la rubrique Chiffrement au repos dans le guide de l'utilisateur ou le guide du développeur du service.

Clés gérées par le client

Les clés KMS que vous créez sont des clés gérées par le client. Les clés gérées par le client sont des clés KMS de votre Compte AWS que vous créez, possédez et gérez. Vous disposez d'un contrôle total sur ces clés KMS, y compris établir et maintenir leurs politiques de clé, les politiques IAM et les octrois, leur activation et leur désactivation, la rotation de leurs éléments de chiffrement, l'ajout de balises, la création d'alias qui font référence aux clés KMS, et la planification des clés KMS en vue de leur suppression.

Les clés gérées par le client apparaissent sur la page Clés gérées par le client de la AWS Management Console pour AWS KMS. Pour identifier définitivement une clé gérée par le client, utilisez l'DescribeKeyopération. Pour les clés gérées par le client, la valeur du champ `KeyManager` de la réponse `DescribeKey` est `CUSTOMER`.

Vous pouvez utiliser vos clés gérées par le client dans les opérations de chiffrement et auditer leur utilisation dans les journaux AWS CloudTrail. En outre, de nombreux services AWS qui s'intègrent à

AWS KMS vous permettent de spécifier une clé gérée par le client pour protéger les données qu'ils stockent et gèrent pour vous.

Les clés gérées par le client entraînent des frais mensuels et des frais pour une utilisation au-delà de l'offre gratuite. Ils sont comptés par rapport aux quotas AWS KMS pour votre compte. Pour plus d'informations, consultez Tarification AWS Key Management Service et Quotas.

Clés gérées par AWS

Clés gérées par AWS sont des clés KMS de votre compte qui sont créées, gérées et utilisées en votre nom par un service AWS intégré à AWS KMS.

Quelques services AWS vous permettent de choisir Clé gérée par AWS ou une clé gérée par le client pour protéger vos ressources dans ce service. En général, sauf si vous êtes obligé de contrôler la clé de chiffrement qui protège vos ressources, une Clé gérée par AWS est un bon choix. Vous n'êtes pas obligé de créer ou de gérer la clé ou sa stratégie de clé, et il n'y a jamais de frais mensuel pour une Clé gérée par AWS.

Vous pouvez afficher les clés Clés gérées par AWS dans votre compte, afficher leurs politiques de clé et contrôler leur utilisation dans les journaux AWS CloudTrail. Toutefois, vous ne pouvez pas modifier les propriétés de Clés gérées par AWS, utiliser la rotation, modifier leurs politiques de clés ou utiliser la planification pour la suppression. De plus, vous ne pouvez pas utiliser directement les Clés gérées par AWS dans les opérations de chiffrement ; le service qui les crée les utilise en votre nom.

Les Clés gérées par AWS apparaissent sur la page Clés gérées par AWS de la AWS Management Console pour AWS KMS. Par ailleurs, vous pouvez identifier des Clés gérées par AWS au moyen de leurs alias, qui ont le format `aws/service-name`, tel que `aws/redshift`. Pour identifier définitivement unClés gérées par AWS, utilisez l'DescribeKeyopération. Pour les Clés gérées par AWS, la valeur du champ `KeyManager` de la réponse `DescribeKey` est `AWS`.

Toutes les Clés gérées par AWS sont automatiquement soumises à rotation chaque année. Vous ne pouvez pas modifier cette programmation de rotation.

> **Note**
>
> En mai 2022, AWS KMS a modifié le calendrier de rotation pour les Clés gérées par AWS de tous les trois ans (environ 1 095 jours) à tous les ans (environ 365 jours).
> Les nouvelles Clés gérées par AWS sont automatiquement soumises à une rotation un an après leur création, puis environ chaque année par la suite.

Les Clés gérées par AWS existantes sont automatiquement soumises à une rotation un an après leur rotation la plus récente, puis chaque année par la suite.

Il n'y a pas de frais mensuel pour Clés gérées par AWS. Ils peuvent être soumis à des frais pour utilisation en cas de dépassement de l'offre gratuite, mais certains services AWS couvrent ces coûts pour vous. Pour plus de détails, reportez-vous à la rubrique Chiffrement au repos dans le Guide de l'utilisateur ou le guide du développeur du service. Pour plus d'informations, consultez Tarification AWS Key Management Service.

Les Clés gérées par AWS ne sont pas prises en compte dans les quotas de ressources dans le nombre de clés KMS dans chaque région de votre compte. Mais lorsqu'elles sont utilisées pour le compte d'un principal dans votre compte, ces clés KMS sont prises en compte dans les quotas de demandes. Pour plus de détails, consultez Quotas.

Clés détenues par AWS

Les Clés détenues par AWS constituent une collection de clés KMS qu'un service AWS possède et gère pour une utilisation dans plusieurs comptes Comptes AWS. Bien que les Clés détenues par AWS ne soient pas dans votre Compte AWS, un service AWS peut utiliser une Clé détenue par AWS pour protéger les ressources de votre compte.

Quelques services AWS vous permettent de choisir Clé détenue par AWS ou une clé gérée par le client. En général, à moins que vous ne deviez auditer ou contrôler la clé de chiffrement qui protège vos ressources, une Clé détenue par AWS est un bon choix. Clés détenues par AWS sont totalement gratuites (pas de frais mensuel ni de frais d'utilisation), elles ne sont pas pris en compte dans les quotas AWS KMS pour votre compte, et elles sont faciles à utiliser. Vous n'avez pas besoin de créer ou de maintenir la clé ou sa politique de clé.

La rotation des Clés détenues par AWS varie en fonction des services. Pour plus d'informations sur la rotation d'une Clé détenue par AWS donnée, veuillez consulter la rubrique Chiffrement au repos dans le guide de l'utilisateur ou dans le guide du développeur du service.

Clés KMS de chiffrement symétrique

Lorsque vous créez une AWS KMS key, vous obtenez par défaut une clé KMS pour le chiffrement symétrique. Il s'agit du type de clé KMS de base et le plus couramment utilisé.

DansAWS KMS, une clé KMS de chiffrement symétrique représente une clé de chiffrement AES-GCM 256 bits, sauf dans les régions de Chine, où elle représente une clé de chiffrement SM4 128

bits. Les éléments de clé symétrique ne quittent jamais AWS KMS non chiffrés. Pour utiliser une clé KMS de chiffrement symétrique, vous devez appeler AWS KMS. Les clés de chiffrement symétriques sont utilisées dans le chiffrement symétrique, où la même clé est utilisée pour chiffrer et déchiffrer. À moins que votre tâche ne nécessite explicitement un chiffrement asymétrique, les clés KMS de chiffrement symétriques, qui ne quittent jamais AWS KMS non chiffrées, sont un bon choix.

Les services AWS qui sont intégrés à AWS KMS utilisent des clés KMS de chiffrement symétriques pour chiffrer vos données. Ces services ne prennent pas en charge le chiffrement avec des clés KMS asymétriques. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Techniquement, la spécification d'une clé symétrique est SYMMETRIC_DEFAULT, l'utilisation de la clé est ENCRYPT_DECRYPT et l'algorithme de chiffrement est SYMMETRIC_DEFAULT. Pour plus de détails, consultez Spécification de clé SYMMETRIC_DEFAULT.

Vous pouvez utiliser une clé KMS de chiffrement symétrique dans AWS KMS pour chiffrer, déchiffrer et rechiffrer des données et générer des clés de données et des paires de clés. Vous pouvez créer des clés KMS de chiffrement symétriques multi-région, importer vos propres éléments de clé vers une clé KMS de chiffrement symétrique et créer des clés KMS de chiffrement symétriques dans des magasins de clés personnalisés. Pour obtenir un tableau de comparaison des opérations que vous pouvez exécuter sur des clés KMS de différents types, veuillez consulter.Référence des types de clés.

Clés KMS asymétriques

Vous pouvez créer une clé KMS asymétrique dans AWS KMS. Une clé KMS asymétrique représente une paire de clés publiques et de clés privées mathématiquement liées entre elles. La clé privée ne quitte jamais AWS KMS non chiffrée. Pour utiliser la clé privée, vous devez appeler AWS KMS. Vous pouvez utiliser la clé publique dans AWS KMS en appelant des opérations d'API AWS KMS, ou télécharger la clé publique et l'utiliser en dehors de AWS KMS. Vous pouvez également créer des clés KMS multi-région asymétriques.

Vous pouvez créer des clés KMS asymétriques qui représentent des paires de clés RSA ou des paires de clés SM2 (régions chinoises uniquement) pour le chiffrement de clé publique ou la signature et la vérification, ou des paires de clés à courbe elliptique pour la signature et la vérification.

Pour en savoir plus sur la création et l'utilisation des clés KMS asymétriques, veuillez consulter Clés KMS asymétriques dans AWS KMS.

Clés KMS HMAC

Une clé KMS HMAC représente une clé symétrique de longueur variable utilisée pour générer et vérifier les codes d'authentification de message utilisant hash (HMAC). Les éléments d'une clé HMAC ne quittent jamais AWS KMS non chiffrés. Pour utiliser une clé HMAC, appelez les opérations d'API `GenerateMac` ou `VerifyMac`.

Vous pouvez également créer des clés KMS HMAC multi-région.

Pour en savoir plus sur la création et l'utilisation des clés KMS, veuillez consulter Clés HMAC dans AWS KMS.

Clés de données

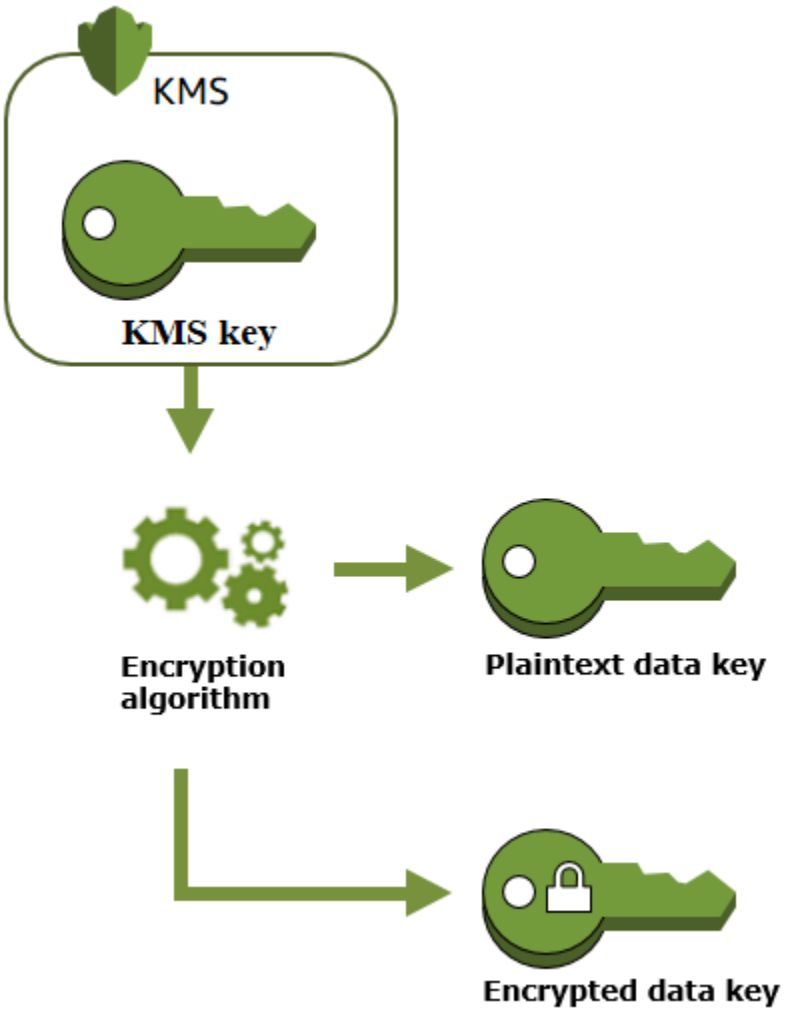
Les clés de données sont des clés symétriques que vous pouvez utiliser pour chiffrer des données, y compris de grandes quantités de données et d'autres clés de chiffrement des données. Contrairement à des clés KMS symétriques, qui ne peuvent pas être téléchargées, les clés de données vous sont renvoyées pour une utilisation en dehors de AWS KMS.

Quand AWS KMS génère des clés de données, il renvoie généralement une clé de données en texte clair pour une utilisation immédiate (facultatif) et une copie chiffrée de la clé de données que vous pouvez stocker en toute sécurité avec les données. Lorsque vous êtes prêt à déchiffrer les données, vous devez d'abord demander à AWS KMS de déchiffrer la clé de données chiffrée.

AWS KMS génère, chiffre et déchiffre des clés de données. Toutefois, AWS KMS ne stocke pas, ne gère pas et ne suit pas vos clés de données, et il n'effectue pas non plus d'opérations cryptographiques avec les clés de données. Vous devez utiliser et gérer les clés de données en dehors d'AWS KMS. Pour obtenir de l'aide sur l'utilisation des clés de données en toute sécurité, consultez AWS Encryption SDK.

Création d'une clé de données

Pour créer une clé de données, appelez l'GenerateDataKeyopération. AWS KMSgénère la clé de données. Il chiffre ensuite une copie de la clé de données sous une clé KMS de chiffrement symétrique que vous spécifiez. Cette opération renvoie une copie en texte clair de la clé de données et la copie de la clé de données chiffrée sous la clé KMS. L'image suivante illustre cette opération.

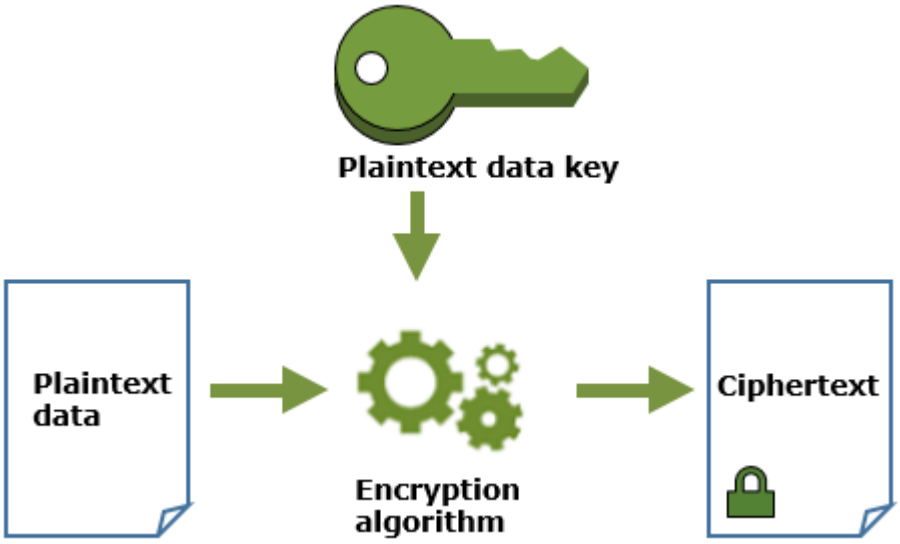


AWS KMSprend également en charge l'GenerateDataKeyWithoutPlaintextopération, qui renvoie uniquement une clé de données cryptée. Lorsque vous avez besoin d'utiliser la clé de données, demandez à AWS KMS de la déchiffrer.

Chiffrement de données avec une clé de données

AWS KMS ne peut pas utiliser une clé de données pour chiffrer les données. Cependant, vous pouvez utiliser la clé de données en dehors de AWS KMS, par exemple en utilisant OpenSSL ou une bibliothèque cryptographique comme AWS Encryption SDK.

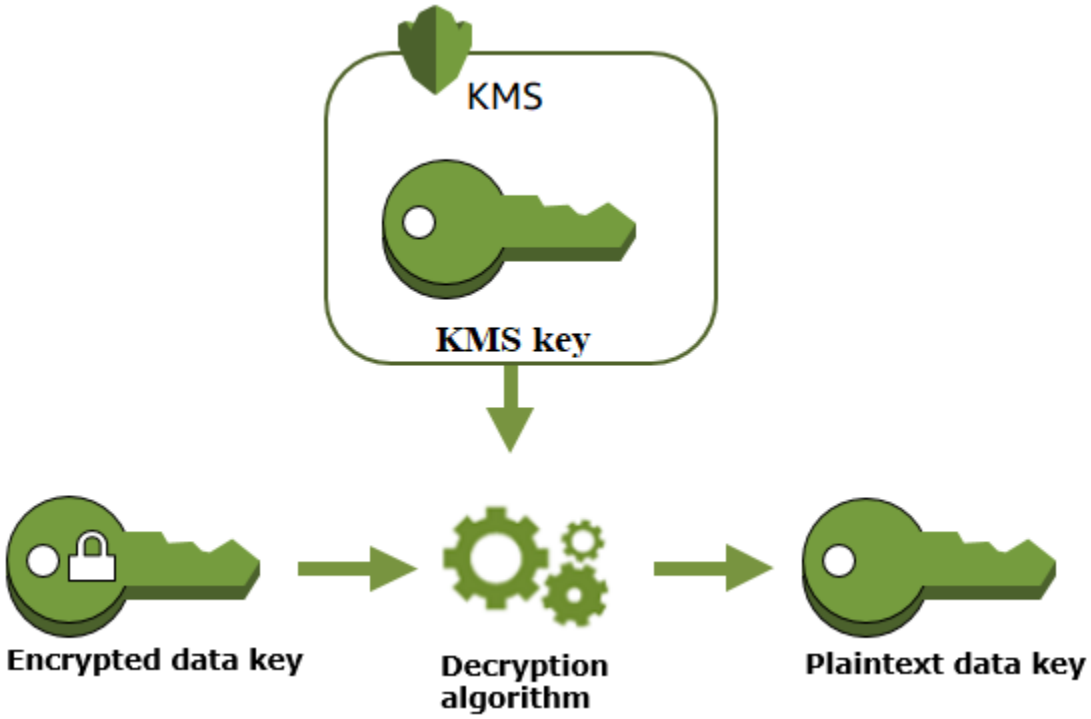
Après avoir utilisé la clé de données en texte brut pour chiffrer les données, supprimez-la de la mémoire dès que possible. Vous pouvez stocker en toute sécurité la clé de données chiffrée avec les données chiffrées pour qu'elle soit disponible pour déchiffrer les données.



Déchiffrement des données avec une clé de données

Pour déchiffrer vos données, transmettez la clé de données chiffrée à l'opération Decrypt. AWS KMS utilise votre clé KMS pour déchiffrer la clé de données, puis renvoie la clé de données en texte brut. Utilisez la clé de données en texte brut pour déchiffrer vos données, puis supprimez la clé de données en texte brut de la mémoire dès que possible.

Le schéma suivant montre comment utiliser l'opération `Decrypt` pour déchiffrer une clé de données chiffrée.



Comment les clés KMS inutilisables affectent les clés de données

Lorsqu'une clé KMS devient inutilisable, l'effet est presque immédiat (sous réserve d'une éventuelle cohérence). L'état de clé de la clé KMS change pour refléter son nouvel état, et toutes les requêtes d'utilisation de la clé KMS dans des opérations cryptographiques échouent.

Cependant, l'effet sur les clés de données chiffrées par la clé KMS, et sur les données chiffrées par la clé de données, est retardé jusqu'à ce que la clé KMS soit utilisée à nouveau, par exemple pour déchiffrer la clé de données.

Les clés KMS peuvent devenir inutilisables pour diverses raisons, notamment les actions suivantes que vous pourriez effectuer.

- Désactiver la clé KMS
- Planifier la suppression de la clé KMS
- Supprimer les éléments de clé d'une clé KMS avec des éléments de clé importés, ou laisser les éléments de clé importés expirer
- Déconnecter le magasin de clés AWS CloudHSM qui héberge la clé KMS ou supprimer la clé du cluster AWS CloudHSM qui fait office d'éléments de clé pour la clé KMS

- Déconnecter le magasin de clés externe qui héberge la clé KMS, ou toute autre action qui interfère avec les requêtes de chiffrement et de déchiffrement adressées au proxy de magasin de clés externe, y compris la suppression de la clé externe de son gestionnaire de clés externe

Cet effet est particulièrement important pour les nombreux Services AWS qui utilisent des clés de données pour protéger les ressources gérées par le service. L'exemple suivant utilise Amazon Elastic Block Store (Amazon EBS) et Amazon Elastic Compute Cloud (Amazon EC2). Différents Services AWS utilisent les clés de données de différentes manières. Pour plus de détails, veuillez consulter la section Protection des données du chapitre Sécurité pour l'Service AWS.

Par exemple, envisagez le scénario suivant :

1. Vous créez un volume EBS chiffré et spécifiez une clé KMS pour le protéger. Amazon EBS demande à AWS KMS d'utiliser votre clé KMS pour générer une clé de données chiffrée pour le volume. Amazon EBS stocke la clé de données chiffrée avec les métadonnées du volume.
2. Lorsque vous attachez le volume EBS à une instance EC2, Amazon EC2 utilise votre clé KMS pour déchiffrer la clé de données chiffrée du volume EBS. Amazon EC2 utilise la clé de données du matériel Nitro, qui est chargé de chiffrer toutes les E/S du disque sur le volume EBS. La clé de données est conservée dans le matériel Nitro tant que le volume EBS est attaché à l'instance EC2.
3. Vous effectuez une action qui rend la clé KMS inutilisable. Cela n'a aucun effet immédiat sur l'instance EC2 ou le volume EBS. Amazon EC2 utilise la clé de données, non pas la clé KMS, pour chiffrer toutes les E/S de disque alors que le volume est attaché à l'instance.
4. Toutefois, lorsque le volume EBS chiffré est détaché de l'instance EC2, Amazon EBS supprime la clé de données en texte brut du matériel Nitro. La prochaine fois que le volume EBS chiffré est attaché à une instance EC2, l'attachement échoue, car Amazon EBS ne peut pas utiliser la clé KMS pour déchiffrer la clé de données chiffrée du volume. Pour utiliser le volume EBS à nouveau, vous devez rendre la clé KMS à nouveau utilisable.

Paires de clés de données

Les paires de clés de données sont des clés de données asymétriques composées d'une clé publique et d'une clé privée mathématiquement liées entre elles. Elles sont conçues pour être utilisées pour le chiffrement et le déchiffrement côté client, ou la signature et la vérification à l'extérieur de AWS KMS.

Contrairement aux paires de clés de données générées par des outils comme OpenSSL, AWS KMS protège la clé privée de chaque paire de clés de données sous une clé KMS de chiffrement

symétrique dans AWS KMS que vous spécifiez. Toutefois, AWS KMS ne stocke pas, ne gère pas et ne suit pas vos paires de clés de données, et il n'effectue pas non plus d'opérations de chiffrement avec les paires de clés de données. Vous devez utiliser et gérer les paires de clés de données en dehors d'AWS KMS.

AWS KMS prend en charge les types de paires de clés de données suivants :

- Paires de clés RSA : RSA_2048, RSA_3072 et RSA_4096
- Paires de clés de courbe elliptique : ECC_NIST_P256, ECC_NIST_P384, ECC_NIST_P521 et ECC_SECG_P256K1
- Paires de clés SM (régions de Chine uniquement) : SM2

Le type de paire de clés de données que vous sélectionnez dépend généralement de votre cas d'utilisation ou des exigences réglementaires. La plupart des certificats nécessitent des clés RSA. Les clés de courbe elliptique sont souvent utilisées pour les signatures numériques. Les clés ECC_SECG_P256K1 sont couramment utilisées pour les cryptomonnaies. AWS KMS recommande d'utiliser des paires de clés ECC pour la signature et des paires de clés RSA pour le chiffrement ou la signature, mais pas pour les deux. Toutefois, AWS KMS ne peut appliquer aucune restriction sur l'utilisation de paires de clés de données en dehors de AWS KMS.

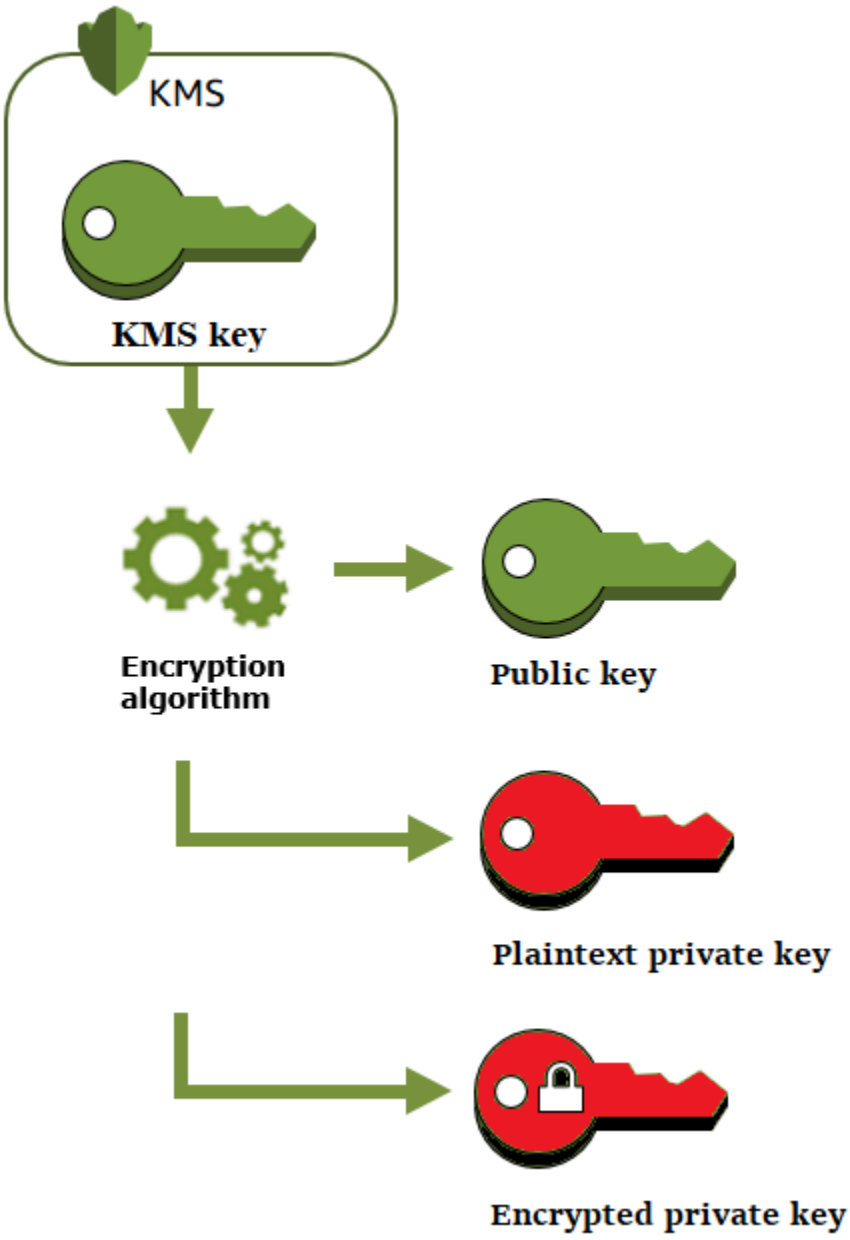
Création d'une paire de clés de données

Pour créer une paire de clés de données, appelez les GenerateDataKeyPairWithoutPlaintextopérations GenerateDataKeyPairor. Spécifiez la clé KMS de chiffrement symétrique que vous souhaitez utiliser pour chiffrer la clé privée.

`GenerateDataKeyPair` renvoie une clé publique en texte brut, une clé privée en texte brut et une clé privée chiffrée. Utilisez cette opération lorsque vous avez besoin immédiatement d'une clé privée en texte brut, par exemple pour générer une signature numérique.

`GenerateDataKeyPairWithoutPlaintext` renvoie une clé publique en texte brut et une clé privée chiffrée, mais pas une clé privée en texte brut. Utilisez cette opération lorsque vous n'avez pas besoin immédiatement d'une clé privée en texte brut, par exemple lorsque vous chiffrez avec une clé publique. Plus tard, lorsque vous avez besoin d'une clé privée en texte brut pour déchiffrer les données, vous pouvez appeler l'opération Decrypt (Déchiffrer).

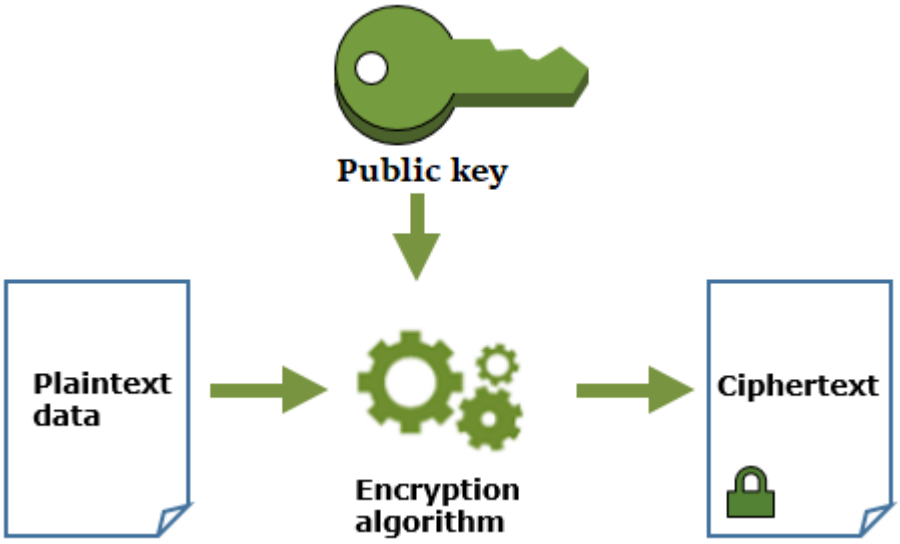
L'image suivante illustre l'opération `GenerateDataKeyPair`. L'opération `GenerateDataKeyPairWithoutPlaintext` omet la clé privée en texte brut.



Chiffrer des données avec une paire de clés de données

Lorsque vous chiffrez avec une paire de clés de données, vous utilisez la clé publique de la paire pour chiffrer les données et la clé privée de la même paire pour déchiffrer les données. Généralement, les paires de clés de données sont utilisées lorsque de nombreuses parties ont besoin de chiffrer des données que seule la partie qui détient la clé privée peut déchiffrer.

Les parties disposant de la clé publique utilisent cette clé pour chiffrer les données, comme indiqué dans le diagramme suivant.

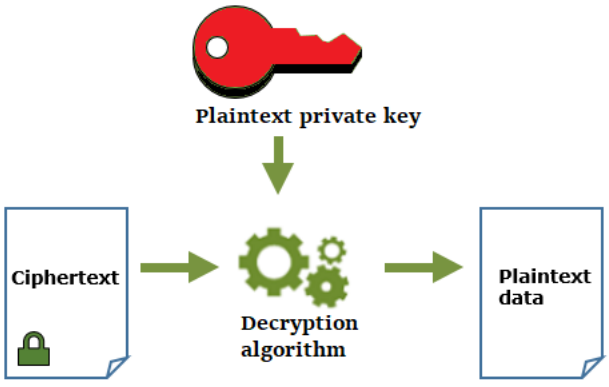


Déchiffrer des données avec une paire de clés de données

Pour déchiffrer vos données, utilisez la clé privée de la paire de clés de données. Pour que l'opération réussisse, les clés publiques et privées doivent être issues de la même paire de clés de données et vous devez utiliser le même algorithme de chiffrement.

Pour déchiffrer la clé privée chiffrée, transmettez-la à l'opération Decrypt (Déchiffrer). Utilisez la clé privée en texte brut pour déchiffrer les données. Ensuite, retirez la clé privée en texte brut de la mémoire dès que possible.

Le diagramme suivant montre comment utiliser la clé privée dans une paire de clés de données pour déchiffrer le texte chiffré.



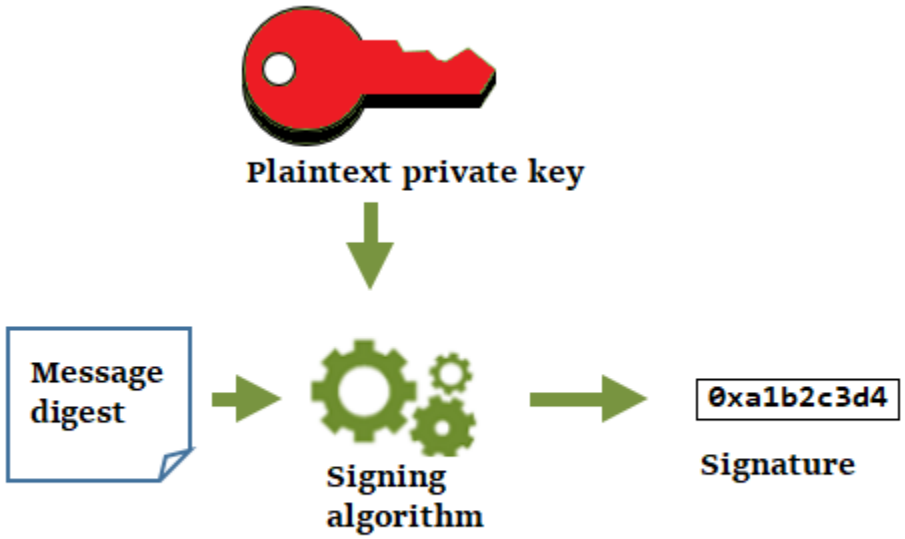
Signer des messages avec une paire de clés de données

Pour générer une signature de chiffrement pour un message, utilisez la clé privée dans la paire de clés de données. Toute personne disposant de la clé publique peut l'utiliser pour vérifier que le message a été signé avec votre clé privée et qu'il n'a pas changé depuis qu'il a été signé.

Si vous chiffrez votre clé privée, transmettez la clé privée chiffrée à l'opération de Decrypt (Déchiffrer). AWS KMS utilise votre clé KMS pour déchiffrer la clé de données, puis il retourne la clé privée en texte brut. Utilisez la clé privée en texte brut pour générer la signature. Ensuite, retirez la clé privée en texte brut de la mémoire dès que possible.

Pour signer un message, créez un résumé de message à l'aide d'une fonction de hachage de chiffrement, telle que la commande `dgst` dans OpenSSL. Ensuite, passez votre clé privée en texte brut à l'algorithme de signature. Le résultat est une signature qui représente le contenu du message. (Vous pourriez être en mesure de signer des messages plus courts sans créer d'abord un résumé. La taille maximale du message varie en fonction de l'outil de signature que vous utilisez.)

Le diagramme suivant montre comment utiliser la clé privée dans une paire de clés de données pour signer un message.



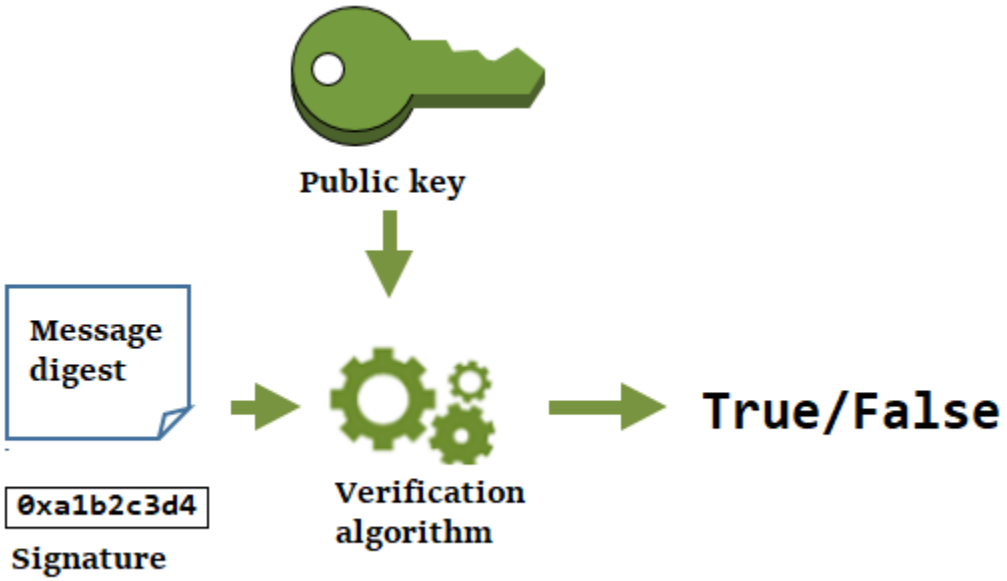
Vérifier une signature avec une paire de clés de données

Toute personne disposant de la clé publique dans votre paire de clés de données peut l'utiliser pour vérifier la signature que vous avez générée avec votre clé privée. La vérification confirme qu'un

utilisateur autorisé a signé le message avec la clé privée et l'algorithme de signature spécifiés, et que le message n'a pas changé depuis sa signature.

Pour réussir, la partie qui vérifie la signature doit générer le même type de résumé, utiliser le même algorithme et utiliser la clé publique qui correspond à la clé privée utilisée pour signer le message.

Le diagramme suivant montre comment utiliser la clé publique dans une paire de clés de données pour vérifier une signature de message.



Alias

Utilisez un alias comme un nom convivial pour une clé KMS. Par exemple, vous pouvez vous référer à une clé KMS en tant que clé de test au lieu de 1234abcd-12ab-34cd-56ef-1234567890ab.

Les alias facilitent l'identification d'une clé KMS dans la AWS Management Console. Vous pouvez utiliser un alias pour identifier une clé KMS dans certaines opérations AWS KMS, y compris des opérations de chiffrement. Dans les applications, vous pouvez utiliser un seul alias pour faire référence à différentes clés KMS dans chaque Région AWS.

Vous pouvez également autoriser et refuser l'accès aux clés KMS en fonction de leurs alias sans modifier les politiques ni gérer les octrois. Cette fonction fait partie de la prise en charge AWS KMS du contrôle d'accès basé sur les attributs (ABAC). Pour plus de détails, consultez ABAC pour AWS KMS.

Dans AWS KMS, les alias sont des ressources indépendantes et non des propriétés d'une clé KMS. Ainsi, vous pouvez ajouter, modifier et supprimer un alias sans affecter la clé KMS associée.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans un nom d’alias. Les alias peuvent apparaître en texte clair dans les CloudTrail journaux et autres sorties.

En savoir plus :

- Pour plus d'informations sur les alias, veuillez consulter Utilisation des alias.
- Pour plus d'informations sur les formats des identificateurs de clé, y compris les alias, veuillez consulter Identifiants clés () KeyId.
- Pour obtenir de l'aide sur la recherche des alias associés à une clé KMS, veuillez consulter Recherche du nom d'alias et de l'ARN d'alias
- Pour obtenir des exemples de création et de gestion des alias dans plusieurs langages de programmation, veuillez consulter Utilisation des alias.

Magasins de clés personnalisés

Un magasin de clés personnalisé est une ressource AWS KMS soutenue par un gestionnaire de clés à l'extérieur d'AWS KMS que vous possédez et gérez. Lorsque vous utilisez une clé KMS dans un magasin de clés personnalisé pour une opération cryptographique, l'opération cryptographique est en fait effectuée dans votre gestionnaire de clés en utilisant ses clés cryptographiques.

AWS KMS prend en charge les magasins de clés AWS CloudHSM soutenus par un cluster AWS CloudHSM et les magasins de clés externes soutenus par un gestionnaire de clés externe à l'extérieur d'AWS.

Pour plus d’informations, consultez Magasins de clés personnalisés.

Opérations cryptographiques

Dans AWS KMS, les opérations de chiffrement sont des opérations d'API qui utilisent des clés KMS pour protéger les données. Comme les clés KMS restent dans AWS KMS, vous devez appeler AWS KMS pour utiliser une clé KMS dans une opération cryptographique.

Pour effectuer des opérations de chiffrement avec des clés KMS, utilisez les kits SDK AWS, AWS Command Line Interface (AWS CLI) ou le AWS Tools for PowerShell. Vous ne pouvez pas effectuer d'opérations cryptographiques dans la console AWS KMS. Pour obtenir des exemples d'appel des opérations cryptographiques dans plusieurs langages de programmation, veuillez consulter Programmation de l'API AWS KMS.

Le tableau suivant répertorie les opérations de chiffrement AWS KMS. Il indique également le type de clé et les exigences d'utilisation des clés KMS utilisées dans l'opération.

Opération	Type de clé	Utilisation de la clé
Decrypt	Symétrique ou asymétrique	`ENCRYPT_DECRYPT`
Encrypt	Symétrique ou asymétrique	`ENCRYPT_DECRYPT`
GenerateDataKey	Symétrique	`ENCRYPT_DECRYPT`
GenerateDataKeyPair	Symétrique [1] Non pris en charge sur les clés KMS dans les magasins de clés personnalisés.	`ENCRYPT_DECRYPT`
GenerateDataKeyPairWithoutPlaintext	Symétrique [1] Non pris en charge sur les clés KMS dans les magasins de clés personnalisés.	`ENCRYPT_DECRYPT`
GenerateDataKeyWithoutPlaintext	Symétrique	`ENCRYPT_DECRYPT`
GenerateMac	HMAC	`GENERATE_VERIFY_MAC`

Opération	Type de clé	Utilisation de la clé
GenerateRandom	N/A. Cette opération n'utilise pas de clé KMS.	N/A
ReEncrypt	Symétrique ou asymétrique	`ENCRYPT_DECRYPT`
Sign (Signer)	Asymétrique	`SIGN_VERIFY`
Vérification	Asymétrique	`SIGN_VERIFY`
VerifyMac	HMAC	`GENERATE_VERIFY_MAC`

[1] Génère une paire de clés de données asymétriques qui est protégée par une clé KMS de chiffrement symétrique.

Pour plus d'informations sur les autorisations pour les opérations cryptographiques, veuillez consulter the section called “Référence des autorisations ”.

Pour rendre AWS KMS réactif et très fonctionnel pour tous les utilisateurs, AWS KMS établit des quotas sur le nombre d'opérations de chiffrement appelées dans chaque seconde. Pour plus de détails, consultez the section called “Quotas partagés pour les opérations de chiffrement”.

Identifiants clés () KeyId

Les identificateurs de clé servent de noms pour vos clés KMS. Ils vous aident à reconnaître vos clés KMS dans la console. Vous les utilisez pour indiquer les clés KMS que vous souhaitez utiliser dans les opérations d'API AWS KMS, les politiques de clés, les politiques IAM et les octrois. Les valeurs de l'identifiant de clé ne sont absolument pas liées au matériel clé associé à la clé KMS.

AWS KMS définit plusieurs identificateurs de clés. Lorsque vous créez une clé KMS, AWS KMS génère un ARN de clé et un ID de clé, qui sont des propriétés de la clé KMS. Lorsque vous créez un alias, AWS KMS génère un ARN d'alias basé sur le nom d'alias que vous définissez. Vous pouvez afficher les identificateurs de clé et d'alias dans AWS Management Console et dans l'API AWS KMS.

Dans la console AWS KMS, vous pouvez afficher et filtrer les clés KMS en fonction de leur ARN de clé, de leur ID de clé ou de leur nom d'alias, et trier par ID de clé et nom d'alias. Pour obtenir de l'aide sur la recherche des identificateurs clés dans la console, veuillez consulter the section called “Recherche de l'ID et de l'ARN d'une clé”.

Dans l'API AWS KMS, les paramètres que vous utilisez pour identifier une clé KMS sont nommés `KeyId` ou une variation, par exemple `TargetKeyId` ou `DestinationKeyId`. Toutefois, les valeurs de ces paramètres ne sont pas limitées aux ID de clé. Certains peuvent prendre n'importe quel identifiant de clé valide. Pour de plus amples informations sur les valeurs de chaque paramètre, veuillez consulter la description du paramètre dans la Référence d'API AWS Key Management Service.

> **Note**
>
> Lorsque vous utilisez l'API AWS KMS, faites attention à l'identificateur de clé que vous utilisez. Différentes API nécessitent des identificateurs de clés différents. En général, utilisez l'identificateur de clé le plus complet et le plus pratique pour votre tâche.

AWS KMS prend en charge les identificateurs de clés suivants.

ARN de clé

L'ARN de clé est l'Amazon Resource Name (ARN) d'une clé KMS. Il s'agit d'un identifiant unique et entièrement qualifié pour la clé KMS. Un ARN de clé inclut le Compte AWS, la région et l'ID de clé. Pour obtenir de l'aide sur la recherche de l'ARN de clé d'une clé KMS, veuillez consulter the section called “Recherche de l'ID et de l'ARN d'une clé”.

Le format d'un ARN de clé est le suivant :

```
arn:<partition>:kms:<region>:<account-id>:key/<key-id>
```

Voici un exemple d'ARN de clé pour une clé KMS de région unique.

```
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

L'élément *`key-id`* des ARN de clé des clés multi-région commencent par le préfixe `mrk-`. Voici un exemple d'ARN de clé pour une clé KMS multi-région.

```
arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab
```

ID de clé

L'ID de clé identifie de manière unique une clé KMS au sein d'un compte et d'une région. Pour obtenir de l'aide sur la recherche de l'ID de clé d'une clé KMS, veuillez consulter the section called “Recherche de l'ID et de l'ARN d'une clé”.

Voici un exemple d'ID de clé pour une clé KMS de région unique.

```
1234abcd-12ab-34cd-56ef-1234567890ab
```

Les ID de clé de clés multi-région commencent par le préfixe `mrk-`. Voici un exemple d'ID de clé pour une clé KMS multi-région.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

ARN d'alias

L'ARN d'alias est l'Amazon Resource Name (ARN) d'un alias AWS KMS. Il s'agit d'un identifiant unique et complet pour l'alias et pour la clé KMS qu'il représente. Un ARN d'alias inclut le Compte AWS, la région et le nom d'alias.

À tout moment, un ARN d'alias identifie une clé KMS particulière. Toutefois, comme vous pouvez modifier la clé KMS associée à l'alias, l'ARN d'alias peut identifier différentes clés KMS à des moments différents. Pour obtenir de l'aide sur la recherche de l'ARN d'alias d'une clé KMS, veuillez consulter Recherche du nom d'alias et de l'ARN d'alias.

Le format d'un ARN d'alias est le suivant :

```
arn:<partition>:kms:<region>:<account-id>:alias/<alias-name>
```

Ce qui suit est l'ARN d'alias pour un `ExampleAlias` fictif.

```
arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias
```

Nom d'alias

Le nom d'alias est une chaîne comportant jusqu'à 256 caractères. Il identifie de manière unique une clé KMS associée au sein d'un compte et d'une région. Dans l'API AWS KMS, les noms d'alias commencent toujours par `alias/`. Pour obtenir de l'aide sur la recherche du nom d'alias d'une clé KMS, veuillez consulter Recherche du nom d'alias et de l'ARN d'alias.

Le format d'un nom d'alias est le suivant :

```
alias/<alias-name>
```

Par exemple :

```
alias/ExampleAlias
```

Le préfixe `aws/` d'un nom d'alias est réservé aux Clés gérées par AWS. Vous ne pouvez pas créer d'alias avec ce préfixe. Par exemple, le nom d'alias de la Clé gérée par AWS pour Amazon Simple Storage Service (Amazon S3) est la suivante.

```
alias/aws/s3
```

Éléments de clé

Les éléments de clé sont la chaîne de bits utilisée dans un algorithme de chiffrement. Les éléments de clé secrète doivent être tenus secrets pour protéger les opérations de chiffrement qui les utilisent. Les éléments de clé publique sont conçus pour être partagés.

Chaque clé KMS inclut une référence à ses éléments de clé dans ses métadonnées. L'origine des éléments de clé des clés KMS de chiffrement symétrique peut varier. Vous pouvez utiliser des éléments clés que AWS KMS génère, les éléments de clé qui sont générés dans le cluster AWS CloudHSM d'un magasin de clés personnalisé ou importer vos propres éléments de clé. Si vous utilisez les éléments de clé AWS KMS pour votre clé KMS de chiffrement symétrique, vous pouvez activer la rotation automatique de vos éléments de clé.

Par défaut, chaque clé KMS possède des éléments de clé uniques. Toutefois, vous pouvez créer un ensemble de clés multi-région avec les mêmes éléments de clé.

Origine des éléments de clé

L'origine des éléments de clé est une propriété de clé KMS qui identifie la source des éléments de clé dans la clé KMS. Vous choisissez l'origine des éléments de clé lorsque vous créez la clé KMS, et vous ne pouvez pas la modifier. La source des éléments de clé affecte les caractéristiques de sécurité, de durabilité, de disponibilité, de latence et de débit de la clé KMS.

Pour trouver l'origine matérielle d'une clé KMS, utilisez l'DescribeKeyopération ou consultez la valeur d'origine dans l'onglet Configuration cryptographique de la page détaillée d'une clé KMS dans la AWS KMS console. Pour obtenir de l'aide, veuillez consulter Affichage des clés.

Les clés KMS peuvent avoir l'une des valeurs d'origine des éléments de clé suivantes.

`AWS_KMS`

AWS KMS crée et gère les éléments de clé pour la clé KMS dans son propre magasin de clés. Il s'agit de la valeur par défaut et de la valeur recommandée pour la plupart des clés KMS.

Pour obtenir de l'aide sur la création des éléments de clé à partir de AWS KMS, veuillez consulter Création de clés.

`EXTERNAL (Import key material)`

La clé KMS comporte des éléments de clé importés. Lorsque vous créez une clé KMS avec une origine d'éléments de clé `External`, la clé KMS n'a pas d'élément de clé. Par la suite, vous pouvez importer des éléments de clé dans la clé KMS. Lorsque vous utilisez des éléments de clé importés, vous devez sécuriser et gérer ces éléments de clé à l'extérieur de AWS KMS, y compris le remplacement des éléments de clé en cas d'expiration. Pour plus de détails, consultez À propos des clés importées.

Pour obtenir de l'aide sur la création d'une clé KMS pour les éléments de clé importés, veuillez consulter Étape 1 : création d'une clé KMS sans élément de clé.

`AWS_CLOUDHSM`

AWS KMS crée les éléments de clé dans le cluster AWS CloudHSM pour votre magasin de clés AWS CloudHSM.

Pour obtenir de l'aide avec la création d'une clé KMS dans un magasin de clés AWS CloudHSM, veuillez consulter la rubrique Créer des clés KMS dans un magasin de clés AWS CloudHSM.

`EXTERNAL_KEY_STORE`

Les éléments de clé consistent en une clé cryptographique dans un gestionnaire de clés externe à l'extérieur d'AWS. Cette origine n'est prise en charge que pour les clés KMS dans un magasin de clés externe.

Pour obtenir de l'aide avec la création d'une clé KMS dans un magasin de clés externe, veuillez consulter la rubrique Créer des clés KMS dans un magasin de clés externe.

Spécifications de la clé

La spécification de la clé est une propriété qui représente la configuration de chiffrement d'une clé. La signification de la spécification de la clé diffère selon le type de clé.

- Clés AWS KMS — Les spécifications de la clé déterminent si la clé KMS est symétrique ou asymétrique. Elles déterminent également le type d'éléments de clé et les algorithmes pris en charge. Vous choisissez la spécification de clé lorsque vous créez la clé KMS et vous ne pouvez pas la modifier. La spécification de clé par défaut, SYMMETRIC_DEFAULT, représente une clé de chiffrement symétrique de 256 bits.

> **Note**
>
> Les `KeySpec` pour une clé KMS étaient appelées `CustomerMasterKeySpec`. Le `CustomerMasterKeySpec` paramètre de l'CreateKeyopération est obsolète. Utilisez plutôt le paramètre `KeySpec`, qui fonctionne de la même manière. Pour éviter d'interrompre les modifications, la réponse des DescribeKeyopérations `CreateKey` et inclut désormais les deux `KeySpec` et les `CustomerMasterKeySpec` membres ayant les mêmes valeurs.

Pour obtenir la liste des spécifications de clés et de l'aide sur le choix d'une spécification de clé, veuillez consulter Sélection des spécifications de la clé. Pour trouver la spécification clé d'une clé KMS, utilisez l'DescribeKeyopération ou consultez l'onglet Configuration cryptographique sur la page détaillée d'une clé KMS dans la AWS KMS console. Pour obtenir de l'aide, veuillez consulter Affichage des clés.

Pour limiter les spécifications clés que les principaux peuvent utiliser lors de la création de clés KMS, utilisez la clé de KeySpec condition kms :. Vous pouvez également utiliser la clé de condition `kms:KeySpec` pour autoriser les principaux à appeler des opérations AWS KMS uniquement

sur des clés KMS avec une spécification de clé particulière. Par exemple, vous pouvez rejeter l'autorisation de planifier la suppression d'une clé KMS avec une spécification de clé `RSA_4096`.

- Clés de données (GenerateDataKey) — La spécification de la clé détermine la longueur d'une clé de données AES.
- Paires de clés de données (GenerateDataKeyPair) — La spécification de la paire de clés détermine le type de contenu clé de la paire de clés de données.

Utilisation de la clé

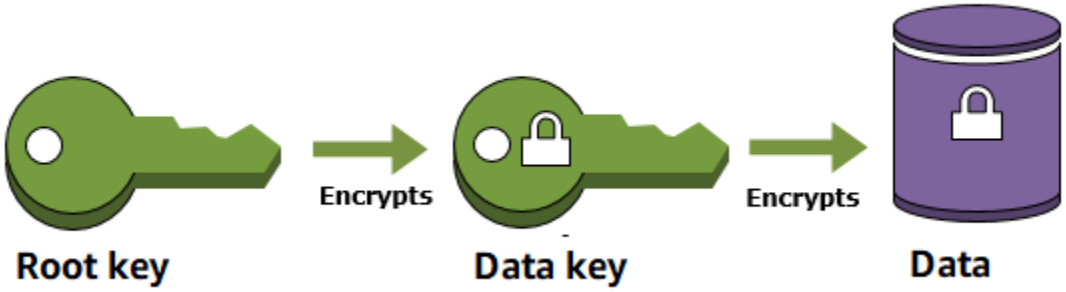
L'utilisation de la clé est une propriété qui détermine les opérations cryptographiques prises en charge par la clé. Les clés KMS peuvent avoir une utilisation de type `ENCRYPT_DECRYPT`, `SIGN_VERIFY` ou `GENERATE_VERIFY_MAC`. Chaque clé KMS ne peut avoir qu'un seul type d'utilisation de clé. L'utilisation d'une clé KMS pour plusieurs types d'opérations rend le produit des deux opérations plus vulnérable aux attaques.

Pour obtenir de l'aide sur le choix de l'utilisation de la clé KMS, veuillez consulter Sélection de l'utilisation des clés. Pour connaître l'utilisation d'une clé KMS, utilisez l'DescribeKeyopération ou choisissez l'onglet Configuration cryptographique sur la page détaillée d'une clé KMS dans la AWS KMS console. Pour obtenir de l'aide, veuillez consulter Affichage des clés.

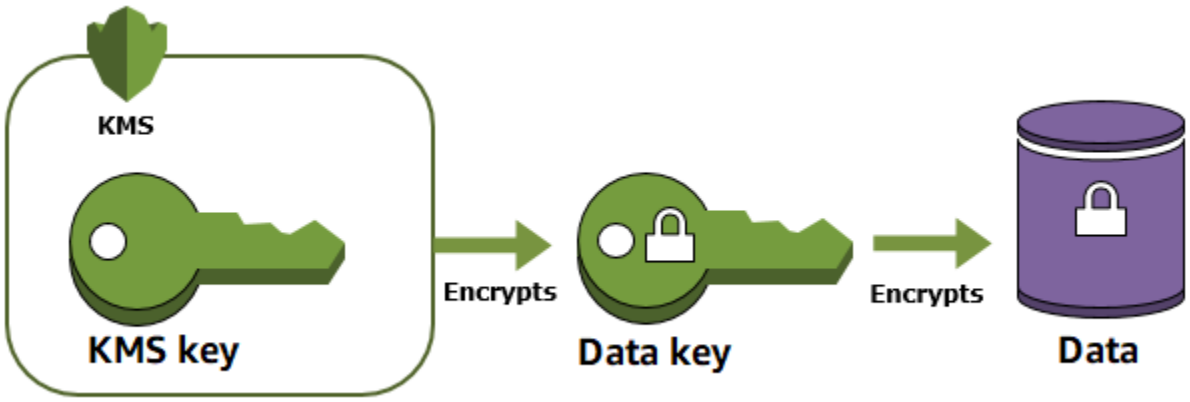
Chiffrement d'enveloppe

Lorsque vous chiffrez vos données, celles-ci sont protégées, mais vous devez protéger votre clé de chiffrement. Une stratégie consiste à la chiffrer. Le chiffrement d'enveloppe est la pratique consistant à chiffrer des données en texte brut à l'aide d'une clé de données, puis à chiffrer la clé de données sous une autre clé.

Vous pouvez même chiffrer la clé de chiffrement de données sous une autre clé de chiffrement et chiffrer cette clé de chiffrement sous une autre clé de chiffrement. Toutefois, au final, une clé doit rester en texte brut pour vous permettre de déchiffrer les clés et vos données. Cette clé de chiffrement de clé en texte brut de niveau supérieur porte le nom de clé racine.



AWS KMS vous aide à protéger vos clés de chiffrement en les stockant et gérant de façon sécurisée. Les clés racine stockées dans AWS KMS, appelées AWS KMS keys, ne quittent jamais les modules de sécurité validés FIPS AWS KMSsans être chiffrées. Pour utiliser une clé KMS, vous devez appeler AWS KMS.



Le chiffrement d'enveloppe offre plusieurs avantages :

- Protection des clés de données

  Lorsque vous chiffrez une clé de données, vous n'avez pas à vous préoccuper du stockage de la clé de données chiffrée, car cette clé de données est intrinsèquement protégée par chiffrement. Vous pouvez stocker en toute sécurité la clé de données chiffrée avec les données chiffrées.
- Chiffrement des mêmes données sous plusieurs clés

  Les opérations de chiffrement peuvent exiger beaucoup de temps, notamment lorsque les données en cours de chiffrement sont des objets de grande taille. Au lieu de rechiffrer des données brutes plusieurs fois avec des clés différentes, vous pouvez rechiffrer uniquement les clés de données qui protègent les données brutes.
- Combinaison des points forts de plusieurs algorithmes

En général, les algorithmes de clé symétrique sont plus rapides et produisent des textes chiffrés plus petits que les algorithmes de clé publique. Cependant, les algorithmes de clé publique fournissent une séparation inhérente des rôles et facilitent la gestion des clés. Le chiffrement d'enveloppe vous permet d'associer les forces de chaque stratégie.

Contexte de chiffrement

Toutes les opérations de chiffrement AWS KMS avec des clés KMS de chiffrement symétriques acceptent un contexte de chiffrement, un ensemble facultatif de paires clé-valeur qui peuvent contenir des informations contextuelles supplémentaires sur les données. AWS KMS utilise le contexte de chiffrement en tant que données authentifiées supplémentaires (AAD) pour prendre en charge le chiffrement authentifié.

Lorsqu'un contexte de chiffrement est inclus dans une requête de chiffrement, il est lié de façon cryptographique au texte chiffré de sorte que le même contexte de chiffrement est requis pour le déchiffrement (ou le déchiffrement et le rechiffrement) des données. Si le contexte de chiffrement fourni dans la requête de déchiffrement ne correspond pas exactement, y compris au niveau des minuscules/majuscules, la requête de déchiffrement échoue. Seul l'ordre des paires clé-valeur dans le contexte de chiffrement peut varier.

> **Note**
>
> Vous ne pouvez pas spécifier de contexte de chiffrement dans une opération de chiffrement avec une clé KMS asymétrique ou une clé KMS HMAC. Les algorithmes asymétriques et les algorithmes MAC ne prennent pas en charge un contexte de chiffrement.

Le contexte de chiffrement n'est pas secret ou chiffré. Il apparaît en texte brut dans les journaux AWS CloudTrail pour vous permettre d'identifier et de classer vos opérations de chiffrement. Votre contexte de chiffrement ne doit pas inclure d'informations sensibles. Nous recommandons que votre chiffrement le contexte décrive les données en cours de chiffrement ou de déchiffrement. Par exemple, lorsque vous chiffrez un fichier, vous pouvez utiliser une partie du chemin de fichier comme contexte de chiffrement.

```
"encryptionContext": {
    "department": "10103.0"
}
```

Par exemple, lors du chiffrement de volumes et d'instantanés créés avec l'opération Amazon Elastic Block Store (Amazon EBS) CreateSnapshot, Amazon EBS utilise l'ID du volume comme valeur de contexte de chiffrement.

```
"encryptionContext": {
  "aws:ebs:id": "vol-abcde12345abc1234"
}
```

Vous pouvez également utiliser le contexte de chiffrement pour affiner ou limiter l'accès aux AWS KMS keys dans votre compte. Vous pouvez utiliser le contexte de chiffrement en tant que contrainte dans les octrois et en tant que condition dans les instructions de politique.

Pour savoir comment utiliser le contexte de chiffrement pour protéger l'intégrité des données chiffrées, consultez le billet Comment protéger l'intégrité de vos données chiffrées en utilisant AWS Key Management Service et EncryptionContext sur le blog sur la AWS sécurité.

En savoir plus sur le contexte de chiffrement.

Règles liées au contexte de chiffrement

AWS KMS applique les règles suivantes pour les valeurs et les clés de contexte de chiffrement.

- La clé et la valeur d'une paire de contexte de chiffrement doivent être des chaînes littérales simples. Si vous utilisez un type différent, tel qu'un nombre entier ou à virgule flottante, AWS KMS l'interprète comme une chaîne.
- Les clés et les valeurs dans un contexte de chiffrement peuvent inclure des caractères Unicode. Si un contexte de chiffrement inclut des caractères non autorisés dans les politiques de clé ou les politiques IAM, vous ne pourrez pas spécifier le contexte de chiffrement dans les clés de condition de politique, telles que `kms:EncryptionContext:context-key` et `kms:EncryptionContextKeys`. Pour plus d'informations sur les règles de document de politique de clé, voir Format de politique de clé. Pour plus d'informations sur les règles du document de politique IAM, veuillez consulter Exigences relatives aux noms IAM dans le Guide de l'utilisateur IAM.

Contexte de chiffrement dans les politiques

Le contexte de chiffrement est utilisé principalement pour vérifier l'intégrité et l'authenticité. Mais vous pouvez également utiliser le contexte de chiffrement pour contrôler l'accès au chiffrement symétrique AWS KMS keys dans les politiques IAM et les politiques clés.

Les clés de EncryptionContextKeys condition kms EncryptionContext : : et kms : autorisent (ou refusent) une autorisation uniquement lorsque la demande inclut des clés de contexte de chiffrement ou des paires clé-valeur particulières.

Par exemple, l'instruction de politique de clé suivante autorise le rôle `RoleForExampleApp` à utiliser la clé KMS dans les opérations `Decrypt`. Elle utilise la clé de condition `kms:EncryptionContext:`*`context-key`* pour accorder cette autorisation uniquement lorsque le contexte de chiffrement de la demande inclut une paire de contexte de chiffrement `AppName:ExampleApp`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

Pour plus d'informations sur ces clés de condition de contexte de chiffrement, consultez Clés de condition pour AWS KMS.

Contexte de chiffrement dans des octrois

Lorsque vous créez un octroi, vous pouvez inclure des contraintes d'octroi qui établissent les conditions des autorisations d'octroi. AWS KMS prend en charge deux contraintes, `EncryptionContextEquals` et `EncryptionContextSubset`, qui impliquent toutes les deux le contexte de chiffrement dans une demande d'opération de chiffrement. Lorsque vous utilisez ces contraintes d'octroi, les autorisations de l'octroi sont effectives uniquement lorsque le contexte de chiffrement de la demande pour l'opération de chiffrement satisfait aux exigences des contraintes d'octroi.

Par exemple, vous pouvez ajouter une contrainte d'`EncryptionContextEquals`autorisation à une autorisation autorisant l'GenerateDataKeyopération. Avec cette contrainte, l'octroi autorise l'opération uniquement lorsque le contexte de chiffrement de la demande est une correspondance sensible à la casse pour le contexte de chiffrement de la contrainte d'octroi.

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/exampleUser \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --operations GenerateDataKey \
    --constraints EncryptionContextEquals={Purpose=Test}
```

Une demande telle que la suivante émanant du principal bénéficiaire satisferait à la contrainte `EncryptionContextEquals`.

```
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --encryption-context Purpose=Test
```

Pour plus d'informations sur les contraintes d'octroi, veuillez consulter Utilisation des contraintes d'octroi. Pour de plus amples informations sur les octrois, veuillez consulter the section called “Octrois”.

Consignation du contexte de chiffrement

AWS KMS utilise AWS CloudTrail pour consigner le contexte de chiffrement et vous permettre de déterminer les clés KMS et les données qui ont été consultées. L'entrée de journal montre exactement quelles clés KMS ont été utilisées pour chiffrer ou déchiffrer les données spécifiques référencées par le contexte de chiffrement dans l'entrée de journal.

> **Important**
>
> Etant donné que le contexte de chiffrement est consigné, il ne doit contenir d'informations sensibles.

Stockage du contexte de chiffrement

Pour simplifier l'utilisation de n'importe quel contexte de chiffrement lorsque vous appelez les opérations `Decrypt` ou `ReEncrypt`, vous pouvez stocker le contexte de chiffrement en même temps que les données chiffrées. Nous vous conseillons de stocker juste assez du contexte de chiffrement pour créer aisément le contexte de chiffrement complet, lorsque cela est nécessaire pour le chiffrement ou le déchiffrement.

Par exemple, si le contexte de chiffrement est le chemin d'accès complet à un fichier, stockez uniquement une partie de ce chemin d'accès avec le contenu du fichier chiffré. Ensuite, lorsque vous aurez besoin du contexte de chiffrement complet, reconstruisez-le à partir du fragment stocké. En cas de tentative d'accès non autorisé au fichier, par exemple pour le renommer ou le déplacer, la valeur du contexte de chiffrement change et la requête de déchiffrement échoue.

Politique de clé

Lorsque vous créez une clé KMS, vous déterminez qui peut utiliser et gérer cette clé KMS. Ces autorisations sont contenues dans un document appelé politique de clé. Vous pouvez utiliser la politique de clé pour ajouter, supprimer ou modifier des autorisations à tout moment pour une clé gérée par le client. Cependant, vous ne pouvez pas modifier la politique de clé pour une Clés gérées par AWS. Pour plus d'informations, consultez Politiques de clé dans AWS KMS.

Octroi

Un octroi est un instrument de politique qui permet aux principaux AWS d'utiliser des AWS KMS keys dans des opérations de chiffrement. Il peut également leur permettre d'afficher une clé KMS (DescribeKey), mais aussi de créer et de gérer des octrois. Lorsque vous autorisez l'accès à une clé KMS, les octrois sont pris en compte avec des politiques de clé et des politiques IAM. Les octrois sont souvent utilisés pour des autorisations temporaires, car vous pouvez en créer un, utiliser ses autorisations et les supprimer sans modifier vos politiques de clé ou IAM. Étant donné que les octrois peuvent être très spécifiques et faciles à créer et à révoquer, ils sont souvent utilisés pour fournir des autorisations temporaires ou des autorisations plus fines.

Pour obtenir des informations détaillées sur les octrois, y compris leur terminologie, veuillez consulter Octrois dans AWS KMS.

Vérification de l'utilisation de clé KMS

Vous pouvez l'utiliser AWS CloudTrail pour auditer l'utilisation des clés. CloudTrail crée des fichiers journaux contenant l'historique des appels d'AWSAPI et des événements associés à votre compte. Ces fichiers journaux incluent toutes les demandes d'API AWS KMS effectuées avec la console de gestion AWS, les kits SDK AWS et les outils de ligne de commande. Les fichiers journaux incluent également les demandes à AWS KMS que les services AWS effectuent en votre nom. Vous pouvez utiliser ces fichiers journaux pour trouver des informations importantes, notamment le moment où les clés KMS ont été utilisées, l'opération qui a été demandée, l'identité du demandeur et l'adresse IP source. Pour plus d'informations, veuillez consulter Journalisation avec AWS CloudTrail et le Guide de l'utilisateur AWS CloudTrail.

Infrastructure de gestion des clés

Une pratique courante dans le domaine du chiffrement consiste à chiffrer et déchiffrer à l'aide d'un algorithme disponible publiquement et évalué par des pairs, tel qu'AES (Advanced Encryption Standard), et d'une clé secrète. L'un des principaux problèmes liés au chiffrement est qu'il est très difficile de maintenir une clé secrète. Il s'agit normalement de la tâche d'une infrastructure de gestion des clés (KMI). AWS KMS gère l'infrastructure de clé pour vous. AWS KMS crée et stocke en toute sécurité vos clés racine, appelées clés AWS KMS keys. Pour plus d'informations sur le fonctionnement de AWS KMS, veuillez consulter Détails cryptographiques deAWS Key Management Service.

Gestion de clés

Pour démarrer avec AWS KMS, créez une AWS KMS key.

Les rubriques de cette section expliquent comment gérer la clé KMS de base, une clé de chiffrement symétrique depuis la création jusqu'à la suppression. Elle contient des rubriques sur l'édition et l'affichage des clés, l'identification des clés, l'activation et la désactivation des clés, la rotation des éléments de clé et l'utilisation des outils et services AWS pour contrôler l'utilisation de vos clés KMS. Elle contient également des informations sur l'utilisation de AWS CloudFormation pour créer et gérer vos clés KMS et une référence d'état de clé qui affiche l'état de clé requis pour chaque opération AWS KMS.

Pour de plus amples informations sur la création, l'utilisation et la gestion d'autres types de clés KMS, veuillez consulter Clés à usage spécial.

Rubriques

- Création de clés
- Utilisation des alias
- Affichage des clés
- Modification des clés
- Clés de balisage
- Activation et désactivation des clés
- Rotation des AWS KMS keys
- Surveillance des AWS KMS keys
- Création de ressources AWS KMS avec AWS CloudFormation
- Suppression de AWS KMS keys
- Principaux états des clés AWS KMS

Création de clés

Vous pouvez créer AWS KMS keys dans ou AWS Management Console en utilisant l'CreateKeyopération ou un AWS CloudFormationmodèle. Au cours de ce processus, vous choisissez le type de clé KMS, de région (région unique ou multiple), ainsi que l'origine des éléments de clé (par défaut, AWS KMS crée les éléments de clé). Vous ne pouvez pas modifier ces propriétés après

la création de la clé KMS. Vous définissez également la politique de clé pour la clé KMS, que vous pouvez modifier à tout moment.

Cette rubrique explique la création d'une clé KMS de base, une clé KMS de chiffrement symétrique pour une seule région avec des éléments de clé de AWS KMS. Vous pouvez utiliser cette clé KMS pour protéger vos ressources dans un Service AWS. Pour plus d'informations sur les clés KMS de chiffrement symétriques, veuillez consulter Spécification de clé SYMMETRIC_DEFAULT. Pour obtenir de l'aide sur la création d'autres types de clés, veuillez consulter Clés à usage spécial.

Si vous créez une clé KMS pour chiffrer les données que vous stockez ou gérez dans un service AWS, créez une clé KMS de chiffrement symétrique. Les services AWS qui sont intégrés à AWS KMS utilisent des clés KMS symétriques pour chiffrer vos données. Ces services ne prennent pas en charge le chiffrement avec des clés KMS asymétriques. Pour obtenir de l'aide sur le choix du type de clé KMS à créer, veuillez consulter Choix d'un type de clé KMS.

> **Note**
>
> Les clés KMS symétriques sont désormais appelées clés KMS de chiffrement symétrique. AWS KMS prend en charge deux types de clés KMS symétriques, les clés KMS de chiffrement symétriques (type par défaut) et les clés KMS HMAC, qui sont également des clés symétriques.

Lorsque vous créez une clé KMS dans la console AWS KMS, vous devez lui donner un alias (nom convivial). L'opération `CreateKey` ne crée pas d'alias pour la nouvelle clé KMS. Pour créer un alias pour une clé KMS nouvelle ou existante, utilisez l'CreateAliasopération. Pour de plus amples informations sur les alias dans AWS KMS, veuillez consulter Utilisation des alias.

Cette rubrique explique la création d'une clé KMS de chiffrement symétrique. Utilisez le tableau suivant pour trouver les instructions relatives à la création de clés KMS de différents types.

Instructions pour créer une clé KMS

Type de clé KMS	Instructions
Clé de chiffrement symétrique (SYMMETRIC_DEFAULT)	the section called “Création de clés KMS de chiffrement symétriques”
Clé asymétrique	the section called “Création de clés KMS asymétriques”

Type de clé KMS	Instructions
Clé HMAC	the section called “Création de clés HMAC”
Clé multi-région (de n'importe quel type)	the section called “Création d'une clé principale avec des éléments de clé importés” the section called “Création d'une clé de réplica avec des éléments de clé importés”
Élément de clé importé (« Bring your own key – BYOK »)	the section called “Étape 1 : création d'une clé KMS sans élément de clé”
Magasin de clés AWS CloudHSM	the section called “Créer des clés KMS dans un magasin de clés AWS CloudHSM”
Magasin de clés externe (« Hold your own key – HYOK »)	the section called “Créer des clés KMS dans un magasin de clés externe”

En savoir plus :

- Pour créer des clés de données pour le chiffrement côté client, utilisez l'GenerateDataKeyopération.
- Pour créer une clé KMS asymétrique à des fins de chiffrement ou de signature, consultez Création de clés KMS asymétriques.
- Pour créer une clé KMS HMAC, veuillez consulter Création de clés KMS HMAC.
- Pour créer une clé KMS avec des éléments de clé importés (« apportez votre propre clé »), veuillez consulter Importation des éléments de clé Étape 1 : créer une AWS KMS key sans élément de clé.
- Pour créer une clé primaire multi-région ou une clé de réplica, consultez Création de clés multi-régions.
- Pour créer une clé KMS dans un magasin de clés personnalisé (l'origine des éléments de clé est le magasin de clés personnalisé (CloudHSM)), veuillez consulter Créer des clés KMS dans un magasin de clés AWS CloudHSM.
- Pour utiliser un AWS CloudFormation modèle pour créer une clé KMS, consultez AWS::KMS::Keyle guide de AWS CloudFormation l'utilisateur.

- Pour déterminer si une clé KMS existante est symétrique ou asymétrique, reportez-vous à Identification des clés KMS asymétriques.
- Pour utiliser vos clés KMS par programmation et dans les opérations d'interface de ligne de commande, vous avez besoin d'un ID de clé ou d'un ARN de clé. Pour obtenir des instructions complètes, veuillez consulter Recherche de l'ID et de l'ARN d'une clé.
- Pour de plus amples informations sur les quotas qui s'appliquent aux clés KMS, veuillez consulter Quotas.

Rubriques

- Autorisations de création de clés KMS
- Création de clés KMS de chiffrement symétriques

Autorisations de création de clés KMS

Pour créer une clé KMS dans la console ou à l'aide des API, vous devez disposer de l'autorisation suivante dans une politique IAM. Dans la mesure du possible, utilisez des clés de condition pour limiter les autorisations. Par exemple, vous pouvez utiliser la clé de KeySpec condition kms : dans une politique IAM pour permettre aux principaux de créer uniquement des clés de chiffrement symétriques.

Pour obtenir un exemple de politique IAM pour les entités qui créent des clés, veuillez consulter Autoriser un utilisateur à créer des clés KMS.

> **Note**
>
> Soyez prudent lorsque vous autorisez les principaux à gérer les balises et les alias. La modification d'une balise ou d'un alias permet d'accorder ou de refuser l'autorisation d'utiliser la clé gérée par le client. Pour plus de détails, veuillez consulter ABAC pour AWS KMS.

- kms : CreateKey obligatoire.
- kms : CreateAlias est nécessaire pour créer une clé KMS dans la console où un alias est requis pour chaque nouvelle clé KMS.
- kms : TagResource est obligatoire pour ajouter des balises lors de la création de la clé KMS.
- iam : CreateServiceLinkedRole est nécessaire pour créer des clés primaires multirégionales. Pour plus de détails, consultez Contrôle de l'accès aux clés multi-régions.

L'PutKeyPolicyautorisation kms : n'est pas requise pour créer la clé KMS. L'autorisation `kms:CreateKey` inclut l'autorisation de définir la politique de clé initiale. Toutefois, vous devez ajouter cette autorisation à la politique de clé lors de la création de la clé KMS pour vous assurer que vous pouvez contrôler l'accès à la clé KMS. L'alternative consiste à utiliser le BypassLockoutSafetyCheckparamètre, ce qui n'est pas recommandé.

Les clés KMS appartiennent au compte AWS dans lequel elles ont été créées. L'utilisateur IAM qui crée une clé KMS n'est pas considéré comme le propriétaire de la clé et il n'est pas automatiquement autorisé à utiliser ou à gérer la clé KMS qu'il a créée. Comme tout autre principal, le créateur de clé doit obtenir l'autorisation via une politique de clé, une politique IAM ou une autorisation. Toutefois, les principaux qui disposent de l'autorisation `kms:CreateKey` peuvent définir la politique de clé initiale et s'octroyer l'autorisation d'utiliser ou de gérer la clé.

Création de clés KMS de chiffrement symétriques

Vous pouvez créer des clés KMS dans la AWS Management Console ou à l'aide de l'API AWS KMS.

Cette rubrique explique la création d'une clé KMS de base, une clé KMS de chiffrement symétrique pour une seule région avec des éléments de clé de AWS KMS. Vous pouvez utiliser cette clé KMS pour protéger vos ressources dans un Service AWS. Pour obtenir de l'aide sur la création d'autres types de clés, veuillez consulter Clés à usage spécial.

Création de clés KMS de chiffrement symétriques (console)

Vous pouvez utiliser la AWS Management Console pour créer des AWS KMS keys (clés KMS).

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key.

5. Pour créer une clé KMS de chiffrement symétrique, pour Key type (Type de clé), choisissez Symmetric (Symétrique).

   Pour de plus amples informations sur la création d'une clé KMS asymétrique dans la console AWS KMS, veuillez consulter Création de clés KMS asymétriques (console).

6. Dans Key usage (Utilisation de la clé), l'option Encrypt and decrypt (Chiffrer et déchiffrer) est sélectionnée pour vous.

   Pour plus d'informations sur la création de clés KMS qui génèrent et vérifient des codes MAC, veuillez consulter Création de clés KMS HMAC.

7. Choisissez Suivant.

   Pour de plus amples informations sur les options avancées, veuillez consulter Clés à usage spécial.

8. Saisissez un alias pour la clé KMS. Le nom d'alias ne peut pas commencer par **`aws/`**. Le préfixe **`aws/`** est réservé par Amazon Web Services pour représenter Clés gérées par AWS dans votre compte.

   > **Note**
   >
   > L'ajout, la suppression ou la mise à jour d'un alias peut permettre d'accorder ou de refuser l'autorisation d'utiliser la clé KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation d'alias pour contrôler l'accès aux clés KMS.

   Un alias est un nom d'affichage que vous pouvez utiliser pour identifier facilement la clé KMS. Nous vous conseillons de choisir un alias qui indique le type de données que vous envisagez de protéger ou l'application que vous pensez utiliser avec la clé KMS.

   Les alias sont requis lorsque vous créez une clé KMS dans la AWS Management Console. Ils sont facultatifs lorsque vous utilisez l'CreateKeyopération.

9. (Facultatif) Saisissez une description pour la clé KMS.

   Vous pouvez ajouter une description maintenant ou la mettre à jour à tout moment, sauf si l'état de la clé est `Pending Deletion` ou `Pending Replica Deletion`. Pour ajouter, modifier ou supprimer la description d'une clé gérée par le client existante, modifiez la description dans l'opération AWS Management Console ou utilisez l'UpdateKeyDescriptionopération.

10. (Facultatif) Saisissez une clé de balise et une valeur de balise facultative. Pour ajouter plus d'une balise à la clé KMS, sélectionnez Add tag (Ajouter une balise).

    > **Note**
    >
    > L'étiquetage ou le désétiquetage d'une clé KMS permet d'accorder ou refuser l'autorisation d'utilisation de cette clé KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation de balises pour contrôler l'accès aux clés KMS.

    Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour de plus amples informations sur l'étiquetage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.

11. Choisissez Suivant.
12. Sélectionnez les utilisateurs et les rôles IAM qui peuvent administrer la clé KMS.

    > **Note**
    >
    > Cette politique de clé donne au Compte AWS le contrôle total de cette clé KMS. Il permet aux administrateurs de compte d'utiliser des politiques IAM pour autoriser d'autres principals à gérer la clé KMS. Pour plus de détails, consultez the section called “politique de clé par défaut”.
    >
    > Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

13. (Facultatif) Pour empêcher les utilisateurs et les rôles IAM sélectionnés de supprimer cette clé KMS, dans la section Key deletion (Suppression de la clé) en bas de la page, décochez la case Allow key administrators to delete this key (Autoriser les administrateurs de clé à supprimer cette clé).
14. Choisissez Suivant.

15. Sélectionnez les utilisateurs et les rôles IAM qui peuvent utiliser la clé dans les opérations de chiffrement.

> **Note**
>
> Cette politique de clé donne au Compte AWS le contrôle total de cette clé KMS. Il permet aux administrateurs de compte d'utiliser des politiques IAM pour autoriser d'autres principals à utiliser la clé KMS dans les opérations de chiffrement. Pour plus de détails, consultez the section called “politique de clé par défaut”.
>
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

16. (Facultatif) Vous pouvez autoriser d'autres Comptes AWS à utiliser cette clé KMS pour les opérations cryptographiques. Pour cela, dans la section Autres Comptes AWS en bas de la page, sélectionnez Ajouter un autre Compte AWS et saisissez le numéro d'identification Compte AWS d'un compte externe. Pour ajouter plusieurs comptes externes, répétez cette étape.

> **Note**
>
> Pour autoriser les principaux des comptes externes à utiliser la clé KMS, les administrateurs du compte externe doivent créer des politiques IAM qui fournissent ces autorisations. Pour plus d’informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

17. Choisissez Suivant.
18. Passez en revue les paramètres de clé que vous avez choisis. Vous pouvez toujours revenir en arrière et modifier tous les paramètres.
19. Choisissez Finish (Terminer) pour créer la clé KMS.

Création de clés KMS de chiffrement symétriques (API AWS KMS)

Vous pouvez utiliser cette CreateKeyopération pour créer tous les AWS KMS keys types. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans les champs `Description` or `Tags`. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

L'opération suivante crée la clé KMS la plus couramment utilisée, une clé de chiffrement symétrique dans une seule région soutenue par les éléments de clé générés par AWS KMS. Cette opération n'a aucun paramètre obligatoire. Vous pouvez également souhaiter utiliser le paramètre `Policy` pour spécifier une politique de clé. Vous pouvez modifier la politique clé (PutKeyPolicy) et ajouter des éléments facultatifs, tels qu'une description et des balises à tout moment. Vous pouvez également créer des clés asymétriques, des clés multi-régions, des clés avec des éléments de clé importés et des clés dans des magasins de clés personnalisés.

L'`CreateKey`opération ne vous permet pas de spécifier un alias, mais vous pouvez l'CreateAliasutiliser pour créer un alias pour votre nouvelle clé KMS.

Voici un exemple d'appel à l'opération `CreateKey` sans paramètre. Cette commande utilise toutes les valeurs par défaut. Elle crée une clé KMS de chiffrement symétrique avec les éléments de clé générés par AWS KMS.

```
$ aws kms create-key
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
```

```
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "MultiRegion": false
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
    }
}
```

Si vous ne spécifiez pas de politique de clé pour votre nouvelle clé KMS, la politique de clé par défaut appliquée par `CreateKey` est différente de celle appliquée par la console lorsque vous utilisez cette dernière pour créer une nouvelle clé KMS.

Par exemple, cet appel à l'GetKeyPolicyopération renvoie la politique clé qui `CreateKey` s'applique. Il donne au Compte AWS l'accès à la clé KMS et lui permet de créer des politiques AWS Identity and Access Management (IAM) pour la clé KMS. Pour plus d'informations sur les politiques IAM et les politiques de clé pour les clés KMS, veuillez consulter Authentification et contrôle d'accès pour AWS KMS

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default --output text
{
  "Version" : "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [ {
    "Sid" : "Enable IAM User Permissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

Utilisation des alias

Un alias est un nom convivial pour une AWS KMS key. Par exemple, un alias vous permet de faire référence à une clé KMS en tant que `test-key` au lieu de `1234abcd-12ab-34cd-56ef-1234567890ab`.

Vous pouvez utiliser un alias pour identifier une clé KMS dans la AWS KMS console, dans l'DescribeKeyopération et dans les opérations cryptographiques, telles que Encrypt et. GenerateDataKey Les alias facilitent également la reconnaissance d'une Clé gérée par AWS. Les alias de ces clés KMS sont toujours au format `aws/<service-name>`. Par exemple, l'alias de la Clé gérée par AWS pour Amazon DynamoDB est `aws/dynamodb`. Vous pouvez établir des normes d'alias similaires pour vos projets, par exemple préfacer vos alias avec le nom d'un projet ou d'une catégorie.

Vous pouvez également autoriser et refuser l'accès aux clés KMS en fonction de leurs alias sans modifier les politiques ni gérer les octrois. Cette fonction fait partie de la prise en charge AWS KMS du contrôle d'accès basé sur les attributs (ABAC). Pour plus de détails, consultez Utilisation d'alias pour contrôler l'accès aux clés KMS.

Une grande partie des performances des alias provient de votre capacité à modifier la clé KMS associée à un alias à tout moment. Les alias peuvent faciliter l'écriture et la maintenance de votre code. Par exemple, supposons que vous utilisiez un alias pour faire référence à une clé KMS particulière et que vous souhaitiez modifier la clé KMS. Dans ce cas, il suffit d'associer l'alias à une autre clé KMS. Vous n'avez pas besoin d'apporter de modifications à votre code.

Les alias facilitent également la réutilisation du même code dans différentes Régions AWS. Créez des alias portant le même nom dans plusieurs régions et associez chaque alias à une clé KMS dans sa région. Lorsque le code s'exécute dans chaque région, l'alias fait référence à la clé KMS associée dans cette région. Pour obtenir un exemple, consultez Utilisation d'alias dans vos applications.

Vous pouvez créer un alias pour une clé KMS dans la AWS KMS console, à l'aide de l'CreateAliasAPI ou à l'aide d'un AWS CloudFormationmodèle.

L'API AWS KMS fournit un contrôle total des alias dans chaque compte et région. L'API inclut des opérations permettant de créer un alias (CreateAlias), d'afficher les noms d'alias et les ARN des alias (ListAliases), de modifier la clé KMS associée à un alias (UpdateAlias) et de supprimer un alias (DeleteAlias). Pour obtenir des exemples de gestion des alias dans plusieurs langages de programmation, veuillez consulter the section called “Utilisation des alias”.

Les ressources suivantes peuvent vous aider à en savoir plus :

- Pour plus d'informations sur les identificateurs de clé KMS, y compris les alias, veuillez consulter Identifiants clés () KeyId.
- Pour obtenir de l'aide sur l'utilisation d'un AWS CloudFormation modèle afin de créer un alias pour une clé KMS, consultez AWS::KMS::Aliasle guide de AWS CloudFormation l'utilisateur.

- Pour obtenir de l'aide sur la recherche des alias associés à une clé KMS, veuillez consulter Recherche du nom d'alias et de l'ARN d'alias
- Pour plus d'informations sur les quotas de ressources pour les alias et les quotas de taux pour les opérations d'API liées aux alias, veuillez consulter Quotas.
- Pour obtenir des exemples de création et de gestion des alias dans plusieurs langages de programmation, veuillez consulter Utilisation des alias.

Rubriques

- À propos des alias
- Gestion des alias
- Utilisation d'alias dans vos applications
- Contrôle de l'accès aux alias
- Utilisation d'alias pour contrôler l'accès aux clés KMS
- Recherche d'alias dans les journaux AWS CloudTrail

À propos des alias

Découvrez comment les alias fonctionnent dans AWS KMS.

Un alias est une ressource AWS indépendante.

> Un alias n'est pas une propriété d'une clé KMS. Les actions que vous effectuez sur l'alias n'affectent pas sa clé KMS associée. Vous pouvez créer un alias pour une clé KMS, puis mettre à jour l'alias afin qu'il soit associé à une autre clé KMS. Vous pouvez même supprimer l'alias sans aucun effet sur la clé KMS associée. Toutefois, si vous supprimez une clé KMS, tous les alias associés à cette clé KMS sont supprimés.
>
> Si vous spécifiez un alias comme ressource dans une politique IAM, la politique fait référence à l'alias et non à la clé KMS associée.

Chaque alias a deux formats.

> Lorsque vous créez un alias, vous spécifiez le nom de l'alias. AWS KMS crée l'ARN d'alias pour vous.
>
> - Un ARN d'alias est un Amazon Resource Name (ARN) qui identifie l'alias de façon unique.
>
> ```
> # Alias ARN
> ```

```
arn:aws:kms:us-west-2:111122223333:alias/<alias-name>
```

- Un nom d'alias qui est unique dans le compte et la région. Dans l'API AWS KMS, le nom d'alias est toujours préfixé par `alias/`. Ce préfixe est omis dans la console AWS KMS.

```
# Alias name
alias/<alias-name>
```

Les alias ne sont pas secrets

Les alias peuvent être affichés en texte clair dans les CloudTrail journaux et autres sorties. N'incluez pas d'informations confidentielles ou sensibles dans le nom de l'alias.

Chaque alias est associé à une clé KMS à la fois.

L'alias et sa clé KMS doivent se trouver dans le même compte et la même région.

Vous pouvez associer un alias à n'importe quelle clé gérée par le client dans le même Compte AWS et la même région. Cependant, vous n'êtes pas autorisé à associer un alias à une Clé gérée par AWS.

Par exemple, cette ListAliasessortie indique que l'`test-key`alias est associé à une seule clé KMS cible, qui est représentée par la `TargetKeyId` propriété.

```
{
     "AliasName": "alias/test-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1593622000.191,
     "LastUpdatedDate": 1593622000.191
}
```

Plusieurs alias peuvent être associés à la même clé KMS.

Par exemple, vous pouvez associer les alias `test-key` et `project-key` à la même clé KMS.

```
{
     "AliasName": "alias/test-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1593622000.191,
     "LastUpdatedDate": 1593622000.191
},
{
```

```
    "AliasName": "alias/project-key",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",
    "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1516435200.399,
    "LastUpdatedDate": 1516435200.399
}
```

Un alias doit être unique dans un compte et une région.

Par exemple, vous ne pouvez avoir qu'un seul alias `test-key` dans chaque compte et région. Les alias sont sensibles à la casse, mais les alias qui ne diffèrent que par leur majuscule sont très propices aux erreurs. Vous ne pouvez pas modifier un nom d'alias. Toutefois, vous pouvez supprimer l'alias et créer un alias avec le nom souhaité.

Vous pouvez créer des alias avec le même nom dans des régions différentes.

Par exemple, vous pouvez avoir un alias `finance-key` dans la région USA Est (Virginie du Nord) et un alias `finance-key` en Europe (Francfort). Chaque alias serait associé à une clé KMS dans sa région. Si votre code fait référence à un nom d'alias comme `alias/finance-key`, vous pouvez l'exécuter dans plusieurs régions. Dans chaque région, il utilise une clé KMS différente. Pour plus de détails, consultez Utilisation d'alias dans vos applications.

Vous pouvez modifier la clé KMS associée à un alias

Vous pouvez utiliser cette UpdateAliasopération pour associer un alias à une autre clé KMS. Par exemple, si l'alias `finance-key` est associé à la clé KMS `1234abcd-12ab-34cd-56ef-1234567890ab`, vous pouvez le mettre à jour afin qu'il soit associé à la clé KMS `0987dcba-09fe-87dc-65ba-ab0987654321`.

Toutefois, la clé KMS actuelle et la nouvelle clé KMS doivent être du même type (toutes deux soit symétriques, soit asymétriques, soit HMAC) et avoir la même utilisation de clé (ENCRYPT_DECRYPT or SIGN_VERIFY ou GENERATE_VERIFY_MAC). Cette restriction empêche les erreurs dans le code qui utilise des alias. Si vous devez associer un alias à un autre type de clé et que vous avez atténué les risques, vous pouvez supprimer et recréer l'alias.

Certaines clés KMS n'ont pas d'alias.

Lorsque vous créez une clé KMS dans la console AWS KMS, vous devez lui attribuer un nouvel alias. Toutefois, aucun alias n'est requis lorsque vous utilisez l'CreateKeyopération pour créer une clé KMS. Vous pouvez également utiliser l'UpdateAliasopération pour modifier la clé KMS associée à un alias et l'DeleteAliasopération pour supprimer un alias. Par conséquent, certaines clés KMS peuvent avoir plusieurs alias, tandis que d'autres peuvent n'en avoir aucun.

AWS crée des alias dans votre compte.

AWS crée des alias dans votre compte pour Clés gérées par AWS. Ces alias ont des noms au format `alias/aws/<service-name>`, comme `alias/aws/s3`.

Certains alias AWS n'ont pas de clé KMS. Ces alias prédéfinis sont généralement associés à une Clé gérée par AWS lorsque vous commencez à utiliser le service.

Utiliser des alias pour identifier les clés KMS

Vous pouvez utiliser un nom d'alias ou un ARN d'alias pour identifier une clé KMS dans le cadre d'opérations cryptographiques DescribeKey, et GetPublicKey. (Si la clé KMS est dans un autre Compte AWS, vous devez utiliser son ARN de clé ou ARN d'alias.) Les alias ne sont pas des identificateurs valides pour les clés KMS dans d'autres opérations AWS KMS. Pour de plus amples informations sur les identificateurs de clé pour chaque opération d'API AWS KMS, veuillez consulter les descriptions des paramètres `KeyId` dans la Référence d'API AWS Key Management Service.

Vous ne pouvez pas utiliser un nom d'alias ou un ARN d'alias pour identifier une clé KMS dans une politique IAM. Pour contrôler l'accès à une clé KMS en fonction de ses alias, utilisez les clés de ResourceAliases condition kms : RequestAlias ou kms :. Pour plus de détails, consultez ABAC pour AWS KMS.

Gestion des alias

Les utilisateurs autorisés peuvent créer, afficher et supprimer des alias. Vous pouvez également mettre à jour un alias, c'est-à-dire associer un alias existant avec une autre clé KMS.

Rubriques

- Création d'un alias
- Affichage des alias
- Mise à jour des alias
- Suppression d'un alias

Création d'un alias

Vous pouvez créer des alias dans la console AWS KMS ou en utilisant des opérations d'API AWS KMS.

L'alias doit être une chaîne de 1 à 256 caractères. Il peut contenir uniquement des caractères alphanumériques, des barres obliques (/), des traits de soulignement (_) et des tirets (-). Le nom d'alias d'une clé gérée par le client ne peut commencer par `alias/aws/`. Le préfixe `alias/aws/` est réservé à une Clé gérée par AWS.

Vous pouvez créer un alias pour une nouvelle clé KMS ou pour une clé KMS existante. Vous pouvez ajouter un alias afin qu'une clé KMS particulière soit utilisée dans un projet ou une application.

Créer un alias (console)

Lorsque vous créez une clé KMS dans la console AWS KMS, vous devez créer un alias pour la nouvelle clé KMS. Pour créer un alias pour une clé KMS existante, utilisez l'option Aliases (Alias) sur la page de détails de la clé KMS.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client. Vous ne pouvez pas gérer des alias pour des Clés gérées par AWS ou des Clés détenues par AWS.
4. Dans la table, choisissez l'alias ou l'ID d'une clé KMS. Ensuite, sur la page de détails de la clé KMS, choisissez l'onglet Aliases (Alias).

   Si une clé KMS possède plusieurs alias, la colonne Aliases (Alias) dans la table affiche un alias et un résumé d'alias, tel que (+n plus). Le choix du résumé d'alias vous mène directement à l'onglet Aliases (Alias) dans la page des détails de la clé KMS.
5. Dans l'onglet Aliases (Alias), choisissez Create alias (Créer un alias). Saisissez un nom d'alias et choisissez Create alias (Créer un alias).

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

> **Note**
>
> N'ajoutez pas le préfixe `alias/`. La console l'ajoute pour vous automatiquement. Si vous saisissez `alias/ExampleAlias`, le nom d'alias réel sera `alias/alias/ExampleAlias`.

Créer un alias (API AWS KMS)

Pour créer un alias, utilisez l'CreateAliasopération. Contrairement au processus de création de clés KMS dans la console, l'CreateKeyopération ne crée pas d'alias pour une nouvelle clé KMS.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

Vous pouvez utiliser l'opération `CreateAlias` pour créer un alias pour une nouvelle clé KMS sans alias. Vous pouvez également utiliser l'opération `CreateAlias` pour ajouter un alias à une clé KMS existante ou pour recréer un alias qui a été accidentellement supprimé.

Dans les opérations d'API AWS KMS, le nom d'alias doit commencer par `alias/`, suivi d'un nom, par exemple `alias/ExampleAlias`. L'alias doit être unique dans le compte et la région. Pour rechercher les noms d'alias déjà utilisés, utilisez l'ListAliasesopération. Le nom de l'alias est sensible à la casse.

Le `TargetKeyId` peut avoir n'importe quelle clé gérée par le client dans la même Région AWS. Pour identifier la clé KMS, utilisez son ID de clé ou son ARN de clé. Vous ne pouvez pas utiliser un autre alias.

L'exemple suivant crée l'alias `example-key` et l'associe à la clé KMS spécifiée. Ces exemples utilisent la AWS Command Line Interface (AWS CLI). Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Utilisation des alias.

```
$ aws kms create-alias \
    --alias-name alias/example-key \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

`CreateAlias` ne renvoie aucune sortie. Pour voir le nouvel alias, utilisez l'opération `ListAliases`. Pour plus de détails, consultez Affichage des alias (API AWS KMS).

Affichage des alias

Les alias facilitent la reconnaissance des clés KMS dans la console AWS KMS. Vous pouvez afficher les alias d'une clé KMS dans la AWS KMS console ou en utilisant l'ListAliasesopération. L'DescribeKeyopération, qui renvoie les propriétés d'une clé KMS, n'inclut pas les alias.

Affichage des alias (console)

Les pages Clés gérées par le client et Clés gérées par AWS dans la console AWS KMS affichent l'alias associé à chaque clé KMS. Vous pouvez également rechercher, trier et filtrer des clés KMS en fonction de leurs alias.

L'image suivante de la console AWS KMS affiche les alias sur la page Clés gérées par le client d'un exemple de compte. Comme le montre l'image, certaines clés KMS n'ont pas d'alias.

Si une clé KMS possède plusieurs alias, la colonne Aliases (Alias) affiche un alias et un résumé d'alias, tel que (+n plus). Le résumé d'alias indique le nombre d'alias supplémentaires associés à la clé KMS et les liens à l'affichage de tous les alias de la clé KMS sur l'onglet Aliases (Alias).

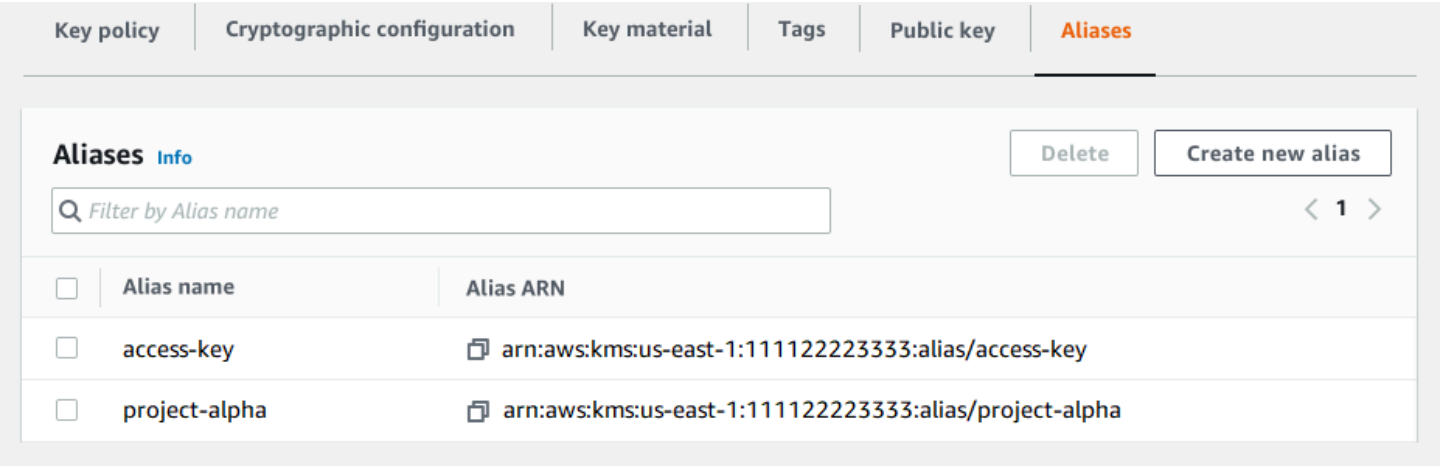
Aliases	Key ID	Status
-	1234abcd-12ab-34cd-56ef-1234567890ab	Enabled
access-key (+1 more)	0987dcba-09fe-87dc-65ba-ab0987654321	Enabled
finance	1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d	Enabled
RSA-4096-Encrypt	1234abcd-09fe-87dc-65ba-5e6f1a2b3c4d	Enabled
RSA-4096-Sign	0987dcba-09fe-34cd-56ef-1234567890ab	Enabled
project-key	1a2b3c4d-5e6f-87dc-65ba-ab0987654321	Enabled

L'onglet Aliases (Alias) de la page de détails de chaque clé KMS affiche le nom d'alias et l'ARN d'alias de tous les alias de la clé KMS dans le Compte AWS et la région. Vous pouvez également utiliser l'onglet Aliases (Alias) pour créer des alias et supprimer des alias.

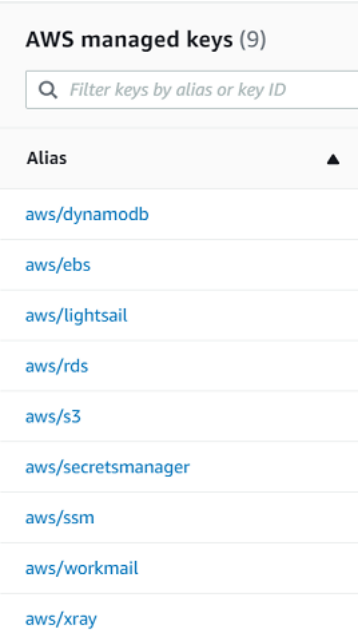
Pour trouver le nom d'alias et l'ARN d'alias de tous les alias de la clé KMS, utilisez l'onglet Aliases (Alias).

- Pour accéder directement à l'onglet Aliases (Alias), dans la colonne Aliases (Alias), choisissez le résumé de l'alias (+n plus). Un résumé d'alias apparaît uniquement si la clé KMS comporte plusieurs alias.
- Vous pouvez également choisir l'alias ou l'ID de clé de la clé KMS (qui ouvre la page des détails de la clé KMS), puis l'onglet Aliases (Alias). Les onglets se trouvent sous la section General configuration (Configuration générale).

L'image suivante illustre l'onglet Aliases (Alias) pour un exemple de clé KMS.



Vous pouvez utiliser l'alias pour reconnaître une Clé gérée par AWS, comme indiqué dans cet exemple de page de Clés gérées par AWS. Les alias de Clés gérées par AWS sont toujours au format `aws/<service-name>`. Par exemple, l'alias de la Clé gérée par AWS pour Amazon DynamoDB est `aws/dynamodb`.



Affichage des alias (API AWS KMS)

L'ListAliasesopération renvoie le nom d'alias et l'ARN d'alias des alias du compte et de la région. La sortie inclut des alias pour Clés gérées par AWS et pour les clés gérées par le client. Les alias de Clés gérées par AWS sont toujours au format `aws/<service-name>`, comme `aws/dynamodb`.

La réponse peut également inclure des alias ne disposant pas de champ `TargetKeyId`. Ces alias prédéfinis ont été créés par AWS qui ne les a pas encore associés à une clé KMS.

```
$ aws kms list-aliases
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
        },
```

```
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
        }
    ]
}
```

Pour obtenir tous les alias associés à une clé KMS spécifique, utilisez le paramètre `KeyId` facultatif de l'opération `ListAliases`. Le paramètre `KeyId` prend l'ID de clé ou l'ARN de clé de la clé KMS.

Cet exemple récupère tous les alias associés à la clé KMS `0987dcba-09fe-87dc-65ba-ab0987654321`.

```
$ aws kms list-aliases --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
```

```
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": "2018-01-20T15:23:10.194000-07:00",
            "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        }
    ]
}
```

Le paramètre `KeyId` ne prend pas de caractères génériques, mais vous pouvez utiliser les fonctions de votre langage de programmation pour filtrer la réponse.

Par exemple, la commande AWS CLI suivante obtient uniquement les alias pour Clés gérées par AWS.

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

Par exemple, la commande suivante obtient uniquement les alias `access-key`. Le nom de l'alias est sensible à la casse.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/access-key`]'
[
    {
        "AliasName": "alias/access-key",
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
        "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": "2018-01-20T15:23:10.194000-07:00",
        "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
    }
]
```

Mise à jour des alias

Étant donné qu'un alias est une ressource indépendante, vous pouvez modifier la clé KMS qui lui est associée. Par exemple, si l'`test-key`alias est associé à une clé KMS, vous pouvez utiliser l'UpdateAliasopération pour l'associer à une autre clé KMS. C'est l'une des nombreuses façons de

faire tourner manuellement une clé KMS sans modifier ses éléments de clé. Vous pouvez également mettre à jour une clé KMS de sorte qu'une application qui utilisait une clé KMS pour de nouvelles ressources utilise désormais une clé KMS différente.

Vous ne pouvez pas mettre à jour un alias dans la console AWS KMS. En outre, vous ne pouvez pas utiliser `UpdateAlias` (ou toute autre opération) pour modifier un nom d'alias. Pour modifier un nom d'alias, supprimez l'alias actuel, puis créez un alias pour la clé KMS.

Lorsque vous mettez à jour un alias, la clé KMS actuelle et la nouvelle clé KMS doivent être du même type (toutes deux soit symétriques, soit asymétriques, soit HMAC). Elles doivent également avoir la même utilisation de la clé (`ENCRYPT_DECRYPT` ou `SIGN_VERIFY` ou GENERATE_VERIFY_MAC). Cette restriction empêche les erreurs cryptographiques dans le code qui utilise des alias.

L'exemple suivant commence par utiliser l'ListAliasesopération pour montrer que l'`test-key`alias est actuellement associé à la clé KMS`1234abcd-12ab-34cd-56ef-1234567890ab`.

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/test-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1593622000.191,
            "LastUpdatedDate": 1593622000.191
        }
    ]
}
```

Ensuite, il utilise l'opération `UpdateAlias` pour modifier la clé KMS associée à l'alias `test-key` avec la clé KMS `0987dcba-09fe-87dc-65ba-ab0987654321`. Vous n'avez pas besoin de spécifier la clé KMS actuellement associée, uniquement la nouvelle clé KMS (« cible »). Le nom de l'alias est sensible à la casse.

```
$ aws kms update-alias --alias-name 'alias/test-key' --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321
```

Pour vérifier que l'alias est maintenant associé à la clé KMS cible, utilisez à nouveau l'opération `ListAliases`. Cette commande AWS CLI utilise le paramètre `--query` pour obtenir uniquement l'alias `test-key`. Les champs `TargetKeyId` et `LastUpdatedDate` sont mis à jour.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[
    {
        "AliasName": "alias/test-key",
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
        "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": 1593622000.191,
        "LastUpdatedDate": 1604958290.154
    }
]
```

Suppression d'un alias

Vous pouvez supprimer un alias dans la AWS KMS console ou en utilisant l'DeleteAliasopération. Avant de supprimer un alias, vérifiez qu'il n'est pas en cours d'utilisation. Bien que la suppression d'un alias n'affecte pas la clé KMS associée, elle peut créer des problèmes pour toute application qui utilise l'alias. Si vous supprimez un alias par erreur, vous pouvez en créer un autre portant le même nom et l'associer à la même clé KMS ou à une clé KMS différente.

Si vous supprimez une clé KMS, tous les alias associés à cette clé KMS sont supprimés.

Supprimer les alias (console)

Pour supprimer un alias dans la console AWS KMS, utilisez l'onglet Aliases (Alias) sur la page de détails de la clé KMS. Vous pouvez supprimer plusieurs alias d'une clé KMS en même temps.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client. Vous ne pouvez pas gérer des alias pour des Clés gérées par AWS ou des Clés détenues par AWS.
4. Dans la table, choisissez l'alias ou l'ID d'une clé KMS. Ensuite, sur la page de détails de la clé KMS, choisissez l'onglet Aliases (Alias).

   Si une clé KMS possède plusieurs alias, la colonne Aliases (Alias) dans la table affiche un alias et un résumé d'alias, tel que (+n plus). Le choix du résumé d'alias vous mène directement à l'onglet Aliases (Alias) dans la page des détails de la clé KMS.

5. Dans l'onglet Aliases (Alias), cochez la case en regard des alias que vous souhaitez supprimer. Ensuite, choisissez Supprimer.

Supprimer un alias (API AWS KMS)

Pour supprimer un alias, utilisez l'DeleteAliasopération. Cette opération supprime un alias à la fois. Le nom de l'alias est sensible à la casse et doit être précédé du préfixe `alias/`.

Par exemple, la commande suivante supprime l'alias `test-key`. Cette commande ne renvoie aucune sortie.

```
$ aws kms delete-alias --alias-name alias/test-key
```

Pour vérifier que l'alias est supprimé, utilisez l'ListAliasesopération. La commande suivante utilise le paramètre `--query` dans la AWS CLI pour obtenir uniquement l'alias `test-key`. Les crochets vides dans la réponse indiquent que la réponse `ListAliases` n'incluait pas d'alias `test-key`. Pour éliminer les crochets, utilisez le paramètre et la valeur `--output text`.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[]
```

Utilisation d'alias dans vos applications

Vous pouvez utiliser un alias pour représenter une clé KMS dans votre code d'application. `KeyId`Paramètre utilisé dans les opérations AWS KMS cryptographiques DescribeKey, et GetPublicKeyaccepte un nom d'alias ou un ARN d'alias.

Par exemple, la commande `GenerateDataKey` suivante utilise un nom d'alias (`alias/finance`) pour identifier une clé KMS. Le nom de l'alias est la valeur du paramètre `KeyId`.

```
$ aws kms generate-data-key --key-id alias/finance --key-spec AES_256
```

Si la clé KMS est dans un autre Compte AWS, vous devez utiliser un ARN de clé ou un ARN d'alias dans ces opérations. Lorsque vous utilisez un ARN d'alias, n'oubliez pas que l'alias d'une clé KMS est défini dans le compte propriétaire de la clé KMS et peut différer d'une région à l'autre. Pour rechercher l'ARN d'alias, veuillez consulter Recherche du nom d'alias et de l'ARN d'alias.

Par exemple, la commande `GenerateDataKey` suivante utilise une clé KMS qui ne se trouve pas dans le compte de l'appelant. L'alias `ExampleAlias` est associé à la clé KMS dans le compte et la région spécifiés.

```
$ aws kms generate-data-key --key-id arn:aws:kms:us-west-2:444455556666:alias/
ExampleAlias --key-spec AES_256
```

L'une des utilisations les plus performantes des alias est au niveau des applications qui s'exécutent dans plusieurs Régions AWS. Par exemple, vous utilisez peut-être une application mondiale qui utilise une Clé KMS asymétriques RSA pour la signature et la vérification.

- Dans la région USA Ouest (Oregon) (us-west-2), vous souhaitez utiliser `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- En Europe (Francfort) (eu-central-1), vous souhaitez utiliser `arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321`.
- Dans la région Asie-Pacifique (Singapour) (ap-southeast-1), vous souhaitez utiliser `arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d`.

Vous pouvez créer une version différente de votre application dans chaque région ou utiliser un dictionnaire ou une instruction switch pour sélectionner la clé KMS appropriée pour chaque région. Toutefois, il est beaucoup plus facile de créer un alias avec le même nom d'alias dans chaque région. Rappelez-vous que ne nom de l'alias est sensible à la casse.

```
aws --region us-west-2 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

aws --region eu-central-1 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321

aws --region ap-southeast-1 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:ap-
southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
```

Ensuite, utilisez l'alias dans votre code. Lorsque votre code s'exécute dans chaque région, l'alias fait référence à sa clé KMS associée dans cette région. Par exemple, ce code appelle l'opération Sign avec un nom d'alias.

```
aws kms sign --key-id alias/new-app \
    --message $message \
    --message-type RAW \
    --signing-algorithm RSASSA_PSS_SHA_384
```

Toutefois, il existe un risque que l'alias soit supprimé ou mis à jour pour être associé à une autre clé KMS. Dans ce cas, les tentatives de l'application pour vérifier les signatures à l'aide du nom d'alias échouent et vous devrez peut-être recréer ou mettre à jour l'alias.

Pour atténuer ce risque, soyez prudent lorsque vous autorisez les principaux à gérer les alias que vous utilisez dans votre application. Pour plus de détails, consultez Contrôle de l'accès aux alias.

Il existe plusieurs autres solutions pour les applications qui chiffrent les données dans plusieurs Régions AWS, y compris le AWS Encryption SDK.

Contrôle de l'accès aux alias

Lorsque vous créez ou modifiez un alias, vous affectez l'alias et sa clé KMS associée. Par conséquent, les principaux qui gèrent les alias doivent avoir l'autorisation d'appeler l'opération d'alias sur l'alias et sur toutes les clés KMS affectées. Vous pouvez fournir ces autorisations à l'aide de politiques de clé, de politiques IAM et d'octrois.

> **Note**
>
> Soyez prudent lorsque vous autorisez les principaux à gérer les balises et les alias. La modification d'une balise ou d'un alias permet d'accorder ou de refuser l'autorisation d'utiliser la clé gérée par le client. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation d'alias pour contrôler l'accès aux clés KMS.

Pour de plus amples informations sur le contrôle de l'accès à toutes les opérations AWS KMS, veuillez consulter Référence des autorisations .

Les autorisations de création et de gestion des alias fonctionnent comme suit.

km : CreateAlias

Pour créer un alias, le principal a besoin des autorisations suivantes pour l'alias et pour la clé KMS associée.

- `kms:CreateAlias` pour l'alias. Fournissez cette autorisation dans une politique IAM attachée au principal autorisé à créer l'alias.

  L'exemple de déclaration de politique suivant spécifie un alias particulier dans l'élément `Resource`. Toutefois, vous pouvez répertorier plusieurs ARN d'alias ou spécifier un modèle d'alias, tel que « test* ». Vous pouvez également spécifier une valeur `Resource` de `"*"` pour permettre au principal de créer un alias dans le compte et la région. L'autorisation de créer un alias peut également être incluse dans une autorisation `kms:Create*` pour toutes les ressources d'un compte et d'une région.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- `kms:CreateAlias` pour la clé KMS. Cette autorisation doit être fournie dans une politique de clé ou dans une IAM politique déléguée par la politique de clé.

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:CreateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Vous pouvez utiliser des clés de condition pour limiter les clés KMS que vous pouvez associer à un alias. Par exemple, vous pouvez utiliser la clé de KeySpec condition kms : pour autoriser le principal à créer des alias uniquement sur des clés KMS asymétriques. Pour obtenir la liste complète des clés de condition que vous pouvez utiliser pour limiter l'autorisation `kms:CreateAlias` sur les ressources de clé KMS, veuillez consulter AWS KMS autorisations.

km : ListAliases

Pour répertorier les alias dans le compte et la région, le principal doit avoir une autorisation `kms:ListAliases` dans une politique IAM. Étant donné que cette politique n'est pas liée à une clé KMS particulière ou à une ressource d'alias, la valeur de l'élément de ressource dans la politique doit être `"*"`.

Par exemple, la déclaration de politique IAM suivante donne au principal l'autorisation de répertorier toutes les clés et tous les alias KMS dans le compte et la région.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

km : UpdateAlias

Pour modifier la clé KMS associée à un alias, le principal a besoin de trois éléments d'autorisation : un pour l'alias, un pour la clé KMS actuelle et un pour la nouvelle clé KMS.

Par exemple, supposons que vous souhaitiez modifier l'alias `test-key` de la clé KMS avec l'ID de clé 1234abcd-12ab-34cd-56ef-1234567890ab vers la clé KMS avec l'ID de clé 0987dcba-09fe-87dc-65ba-ab0987654321. Dans ce cas, incluez des déclarations de politique similaires aux exemples de cette section.

- `kms:UpdateAlias` pour l'alias. Vous fournissez cette autorisation dans une politique IAM associée au principal. La politique IAM suivante spécifie un alias particulier. Toutefois, vous pouvez

répertorier plusieurs ARN d'alias ou spécifier un modèle d'alias, tel que `"test*"`. Vous pouvez également spécifier une valeur `Resource` de `"*"` pour permettre au principal de mettre à jour un alias dans le compte et la région.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:UpdateAlias",
    "kms:ListAliases",
    "kms:ListKeys"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- `kms:UpdateAlias` pour la clé KMS actuellement associée à l'alias. Cette autorisation doit être fournie dans une politique de clé ou dans une IAM politique déléguée par la politique de clé.

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

- `kms:UpdateAlias` pour la clé KMS que l'opération associe à l'alias. Cette autorisation doit être fournie dans une politique de clé ou dans une IAM politique déléguée par la politique de clé.

```
{
  "Sid": "Key policy for 0987dcba-09fe-87dc-65ba-ab0987654321",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Vous pouvez utiliser des clés de condition pour limiter l'une ou l'autre des clés KMS dans une opération `UpdateAlias`. Par exemple, vous pouvez utiliser une clé de ResourceAliases condition kms : pour autoriser le principal à mettre à jour les alias uniquement lorsque la clé KMS cible possède déjà un alias particulier. Pour obtenir la liste complète des clés de condition que vous pouvez utiliser pour limiter l'autorisation `kms:UpdateAlias` sur une ressource de clé KMS, veuillez consulter AWS KMS autorisations.

km : DeleteAlias

Pour supprimer un alias, le principal a besoin d'une autorisation pour l'alias et pour la clé KMS associée.

Comme toujours, vous devez faire preuve de prudence lorsque vous autorisez les principaux à supprimer une ressource. Toutefois, la suppression d'un alias n'a aucun effet sur la clé KMS associée. Bien que cela puisse provoquer un échec dans une application qui s'appuie sur l'alias, si vous supprimez un alias par erreur, vous pouvez le recréer.

- `kms:DeleteAlias` pour l'alias. Fournissez cette autorisation dans une politique IAM attachée au principal autorisé à supprimer l'alias.

  L'exemple de déclaration de politique suivant spécifie un alias dans l'élément `Resource`. Toutefois, vous pouvez répertorier plusieurs ARN d'alias ou spécifier un modèle d'alias, tel que `"test*"`. Vous pouvez également spécifier une valeur `Resource` de `"*"` pour permettre au principal de supprimer tout alias dans le compte et la région.

  ```
  {
    "Sid": "IAMPolicyForAnAlias",
    "Effect": "Allow",
    "Action": [
      "kms:CreateAlias",
      "kms:UpdateAlias",
      "kms:DeleteAlias"
    ],
    "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
  }
  ```

- `kms:DeleteAlias` pour la clé KMS associée. Cette autorisation doit être fournie dans une politique de clé ou dans une IAM politique déléguée par la politique de clé.

  ```
  {
    "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  ```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"
  },
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Limitation des autorisations d'alias

Vous pouvez utiliser des clés de condition pour limiter les autorisations d'alias lorsque la ressource est une clé KMS. Par exemple, la politique IAM suivante autorise les opérations d'alias sur les clés KMS d'un compte et d'une région en particulier. Cependant, il utilise la clé de KeyOrigin condition kms : pour limiter davantage les autorisations aux clés KMS dont le contenu clé provient deAWS KMS.

Pour obtenir la liste complète des clés de condition que vous pouvez utiliser pour limiter l'autorisation d'alias sur une ressource de clé KMS, veuillez consulter AWS KMS autorisations.

```
{
  "Sid": "IAMPolicyKeyPermissions",
  "Effect": "Allow",
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_KMS"
    }
  }
}
```

Vous ne pouvez pas utiliser de clés de condition dans une instruction de politique où la ressource est un alias. Pour limiter les alias qu'un principal peut gérer, utilisez la valeur de l'élément `Resource` de la déclaration de politique IAM qui contrôle l'accès à l'alias. Par exemple, les instructions de politique suivantes permettent au principal de créer, de mettre à jour ou de supprimer un alias dans le Compte AWS et la région, sauf si l'alias commence par `Restricted`.

```
{
  "Sid": "IAMPolicyForAnAliasAllow",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/*"
},
{
  "Sid": "IAMPolicyForAnAliasDeny",
  "Effect": "Deny",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/Restricted*"
}
```

Utilisation d'alias pour contrôler l'accès aux clés KMS

Vous pouvez contrôler l'accès aux clés KMS en fonction des alias associés à la clé KMS. Pour ce faire, utilisez les clés de ResourceAliases condition kms : RequestAlias et kms :. Cette fonction fait partie de la prise en charge AWS KMS du contrôle d'accès basé sur les attributs (ABAC).

La clé de condition `kms:RequestAlias` autorise ou refuse l'accès à une clé KMS en fonction de l'alias dans une requête. La clé de condition `kms:ResourceAliases` autorise ou refuse l'accès à une clé KMS en fonction des alias associés à la clé KMS.

Ces fonctionnalités ne vous permettent pas d'identifier une clé KMS à l'aide d'un alias dans l'élément `resource` d'une déclaration de politique. Lorsqu'un alias est la valeur d'un élément `resource`, la politique s'applique à la ressource d'alias et non à toute clé KMS qui pourrait lui être associée.

> **Note**
>
> Les modifications d'alias et de balise peuvent prendre jusqu'à cinq minutes pour avoir une incidence sur l'autorisation de clé KMS. Les modifications récentes peuvent être visibles dans les opérations d'API avant qu'elles n'affectent l'autorisation.

Lorsque vous utilisez des alias pour contrôler l'accès aux clés KMS, tenez compte des éléments suivants :

- Utilisez des alias pour renforcer les bonnes pratiques de l'accès le moins privilégié. N'accordez aux principaux IAM que les autorisations dont ils ont besoin pour les clés KMS qu'ils doivent utiliser ou gérer. Par exemple, utilisez des alias pour identifier les clés KMS utilisées pour un projet. Donnez ensuite à l'équipe de projet l'autorisation d'utiliser uniquement les clés KMS avec les alias du projet.
- Soyez prudent lorsque vous donnez aux principaux les autorisations `kms:CreateAlias`, `kms:UpdateAlias` ou `kms:DeleteAlias` qui leur permettent d'ajouter, de modifier et de supprimer des alias. Lorsque vous utilisez des alias pour contrôler l'accès aux clés KMS, la modification d'un alias peut donner aux principaux l'autorisation d'utiliser des clés KMS qu'ils n'avaient alors pas l'autorisation d'utiliser. Elle peut également refuser l'accès aux clés KMS dont d'autres principaux ont besoin pour réaliser leurs tâches.
- Examinez les principaux dans votre Compte AWS qui disposent actuellement de l'autorisation de gérer les alias et ajustez les autorisations, si nécessaire. Les administrateurs de clés qui n'ont pas l'autorisation de modifier les politiques de clé ou de créer des octrois peuvent contrôler l'accès aux clés KMS s'ils sont autorisés à gérer les alias.

  Par exemple, la console Politique de clé par défaut pour les administrateurs de clés comprend les autorisations `kms:CreateAlias`, `kms:DeleteAlias` et `kms:UpdateAlias`. Les politiques IAM peuvent donner des autorisations d'alias pour toutes les clés KMS de votre Compte AWS. Par exemple, la politique AWSKeyManagementServicePowerUsergérée permet aux principaux de créer, de supprimer et de répertorier des alias pour toutes les clés KMS, mais pas de les mettre à jour.
- Avant de définir une politique qui dépend d'un alias, examinez les alias des clés KMS dans votre Compte AWS. Assurez-vous que votre politique s'applique uniquement aux alias que vous avez l'intention d'inclure. Utilisez CloudTrail les journaux et les CloudWatch alarmes pour vous avertir des modifications d'alias susceptibles d'affecter l'accès à vos clés KMS. La ListAliasesréponse inclut également la date de création et la date de dernière mise à jour pour chaque alias.

- Les conditions de politique d'alias utilisent la correspondance de modèles ; elles ne sont pas liées à une instance particulière d'un alias. Une politique qui utilise des clés de condition basées sur des alias affecte tous les alias nouveaux et existants qui correspondent au modèle. Si vous supprimez et recréez un alias qui correspond à une condition de politique, la condition s'applique au nouvel alias, comme c'était le cas pour l'ancien.

La clé de condition `kms:RequestAlias` repose sur l'alias spécifié explicitement dans une demande d'opération. La clé de condition `kms:ResourceAliases` dépend des alias associés à une clé KMS, même s'ils n'apparaissent pas dans la demande.

km : RequestAlias

Autoriser ou refuser l'accès à une clé KMS en fonction de l'alias qui identifie la clé KMS dans une demande. Vous pouvez utiliser la clé de RequestAlias condition kms : dans une politique clé ou une politique IAM. Elle s'applique aux opérations qui utilisent un alias pour identifier une clé KMS dans une demande, à savoir les opérations cryptographiques DescribeKey, et GetPublicKey. Il n'est pas valide pour les opérations d'alias, telles que CreateAliasou DeleteAlias.

Dans la clé de condition, spécifiez un Nom d'alias ou un modèle de nom d'alias. Vous ne pouvez pas spécifier d'ARN d'alias.

Par exemple, la déclaration de politique de clé suivante autorise les principaux à utiliser les opérations spécifiées sur la clé KMS. L'autorisation est en vigueur uniquement lorsque la demande utilise un alias qui inclut `alpha` pour identifier la clé KMS.

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/alpha-developer"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:RequestAlias": "alias/*alpha*"
```

```
    }
  }
}
```

L'exemple suivant de demande d'un principal autorisé remplirait la condition. Cependant, une demande qui a utilisé un ID de clé, un ARN de clé ou un alias différent ne remplirait pas la condition, même si ces valeurs identifiaient la même clé KMS.

```
$ aws kms describe-key --key-id "arn:aws:kms:us-west-2:111122223333:alias/project-
alpha"
```

km : ResourceAliases

Autorisez ou refusez l'accès à une clé KMS en fonction des alias associés à la clé KMS, même si l'alias n'est pas utilisé dans une demande. La clé de ResourceAliases condition kms : vous permet de spécifier un alias ou un modèle d'alias, par exemple`alias/test*`, afin que vous puissiez l'utiliser dans une politique IAM pour contrôler l'accès à plusieurs clés KMS dans la même région. Il est valable pour toute opération AWS KMS qui utilise une clé KMS.

Par exemple, la politique IAM suivante permet aux principaux de gérer la rotation automatique des clés sur les clés KMS dans deux Comptes AWS. Toutefois, l'autorisation s'applique uniquement aux clés KMS associées aux alias commençant par `restricted`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:EnableKeyRotation",
        "kms:DisableKeyRotation",
        "kms:GetKeyRotationStatus"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/*",
        "arn:aws:kms:*:444455556666:key/*"
      ],
      "Condition": {
        "ForAnyValue:StringLike": {
          "kms:ResourceAliases": "alias/restricted*"
```

```
        }
      }
    }
  ]
}
```

La condition `kms:ResourceAliases` est une condition de la ressource, pas la demande. Dès lors, une demande qui ne spécifie pas l'alias peut toujours satisfaire la condition.

L'exemple de demande suivant, qui spécifie un alias correspondant, satisfait à la condition.

```
$ aws kms enable-key-rotation --key-id "alias/restricted-project"
```

Toutefois, l'exemple de demande suivant satisfait également la condition, à condition que la clé KMS spécifiée ait un alias qui commence par `restricted`, même si cet alias n'est pas utilisé dans la demande.

```
$ aws kms enable-key-rotation --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

Recherche d'alias dans les journaux AWS CloudTrail

Vous pouvez utiliser un alias pour représenter une AWS KMS key dans une opération d'API AWS KMS. Lorsque vous le faites, l'alias et l'ARN de clé de la clé KMS sont enregistrés dans l'entrée de journal AWS CloudTrail pour l'événement. L'alias apparaît dans le champ `requestParameters`. L'ARN de clé apparaît dans le champ `resources`. Cela se vérifie même lorsqu'un service AWS utilise une Clé gérée par AWS dans votre compte.

Par exemple, la GenerateDataKeydemande suivante utilise l'`project-key`alias pour représenter une clé KMS.

```
$ aws kms generate-data-key --key-id alias/project-key --key-spec AES_256
```

Lorsque cette demande est enregistrée dans le CloudTrail journal, l'entrée du journal inclut à la fois l'alias et l'ARN de la clé KMS réellement utilisée.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDE",
```

```
        "arn": "arn:aws:iam::111122223333:role/ProjectDev",
        "accountId": "111122223333",
        "accessKeyId": "FFHIJ",
        "userName": "example-dev"
    },
    "eventTime": "2020-06-29T23:36:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.205.123.000",
    "userAgent": "aws-cli/1.18.89 Python/3.6.10
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.12",
    "requestParameters": {
        "keyId": "alias/project-key",
        "keySpec": "AES_256"
    },
    "responseElements": null,
    "requestID": "d93f57f5-d4c5-4bab-8139-5a1f7824a363",
    "eventID": "d63001e2-dbc6-4aae-90cb-e5370aca7125",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


Pour plus de détails sur les AWS KMS opérations de journalisation dans CloudTrail les journaux, consultezJournalisation des appels d'API AWS KMS avec AWS CloudTrail.

Affichage des clés

Vous pouvez utiliser la AWS Management Console ou l'API AWS Key Management Service (AWS KMS) pour afficher des AWS KMS keys dans chaque compte et région, y compris les clés KMS que vous gérez et les clés KMS gérées par AWS.

Rubriques

- Affichage de clés KMS dans la console
- Affichage des clés KMS avec l'API
- Affichage de la configuration de chiffrement des clés KMS
- Recherche de l'ID et de l'ARN d'une clé
- Recherche du nom d'alias et de l'ARN d'alias

Affichage de clés KMS dans la console

Dans la AWS Management Console, vous pouvez afficher des listes de vos clés KMS dans le compte et la région et les détails de chaque clé KMS.

> **Note**
>
> La console AWS KMS affiche les clés KMS dont vous disposez autorisation d'afficher dans votre compte et votre région. Clés KMS dans d'autres Comptes AWS n'apparaissent pas dans la console, même si vous avez l'autorisation de les afficher, les gérer et les utiliser. Pour afficher les clés KMS dans d'autres comptes, utilisez l'DescribeKeyopération.

Rubriques

- Navigation vers les tables de clés
- Naviguer vers les détails de clé
- Tri et filtrage de vos clés KMS
- Affichage des détails de clé KMS
- Personnalisation de vos tables de clés KMS

Navigation vers les tables de clés

Les AWS KMS keys de chaque compte et région sont affichées dans des tables. Il existe des tables distinctes pour les clés KMS que vous créez et les clés que les services AWS créent pour vous.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.

3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client. Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS. Pour plus d'informations sur les différents types de clés KMS, veuillez consulter AWS KMS keys.

> **Tip**
>
> Pour afficher les Clés gérées par AWS auxquelles il manque un alias, utilisez la page Clés gérées par le client.
> La console AWS KMS affiche également les magasins de clés personnalisés dans le compte et la région. Les clés KMS que vous créez dans les magasins de clés personnalisés apparaissent sur la page Customer managed keys (Clés gérées par le client). Pour de plus amples informations sur les magasins de clés personnalisés, veuillez consulter Magasins de clés personnalisés.

Naviguer vers les détails de clé

Il existe une page de détails pour chaque AWS KMS key dans le compte et la région. La page de détails affiche la section Configuration générale de la clé KMS et comprend des onglets qui permettent aux utilisateurs autorisés d'afficher et de gérer la configuration de chiffrement et la politique de clé pour la clé. En fonction du type de clé, la page de détails peut également inclure les onglets Aliases (Alias), Key material (Éléments de clé), Key rotation (Rotation de clé), Public Key (Clé publique), Regionality (Régionalité),) et Tags (Balises).

Pour accéder à la page des détails de clé d'une clé KMS.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client. Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS. Pour plus d'informations sur les différents types de clés KMS, veuillez consulter AWS KMS key.
4. Pour ouvrir la page des détails de clé, dans la table de clés, choisissez l'ID de clé ou l'alias de la clé KMS.

Si la clé KMS comporte plusieurs alias, un résumé d'alias (+n plus) apparaît en regard du nom de l'un des alias. Le choix du résumé d'alias vous mène directement à l'onglet Aliases (Alias) dans la page des détails de la clé.

Tri et filtrage de vos clés KMS

Pour faciliter la recherche de vos clés KMS dans la console, vous pouvez trier et filtrer les tables de clés.

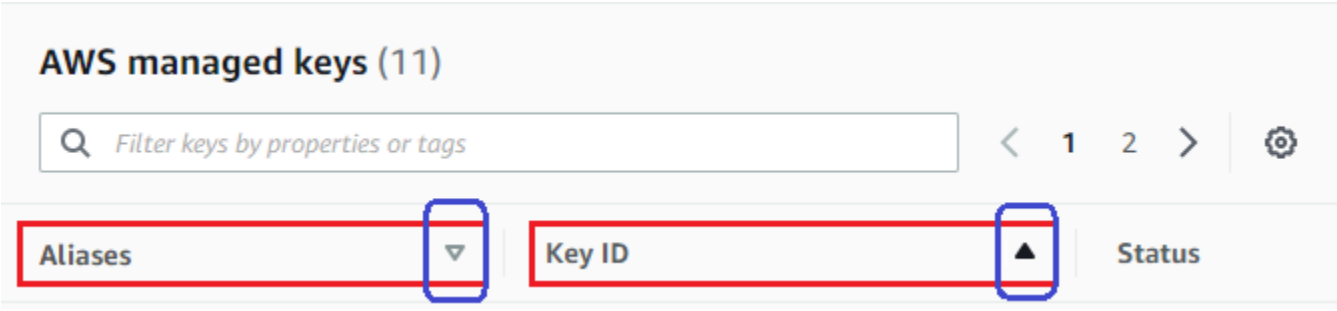
Tri

Vous pouvez trier les clés KMS par ordre croissant ou décroissant selon les valeurs de leurs colonnes. Cette fonction trie toutes les clés KMS de la table, même si elles n'apparaissent pas sur la page de la table en cours.

Les colonnes pouvant être triées sont indiquées par une flèche à côté du nom de la colonne. Dans la page Clés gérées par AWS, vous pouvez trier par Alias ou ID de clé. Sur la page Customer managed keys (Clés gérées par le client), vous pouvez trier par Alias, ID de clé ou Type de clé.

Pour trier par ordre croissant, choisissez l'en-tête de colonne jusqu'à ce que la flèche pointe vers le haut. Pour trier par ordre décroissant, choisissez l'en-tête de colonne jusqu'à ce que la flèche pointe vers le bas. Vous pouvez trier uniquement selon une colonne à la fois.

Par exemple, vous pouvez trier les clés KMS par ordre croissant par ID de clé, au lieu d'alias, qui est la valeur par défaut.



Lorsque vous triez vos clés KMS sur la page Clés gérées par le client dans l'ordre croissant par Type de clé, toutes les clés asymétriques sont affichées avant toutes les clés symétriques.

Filtre

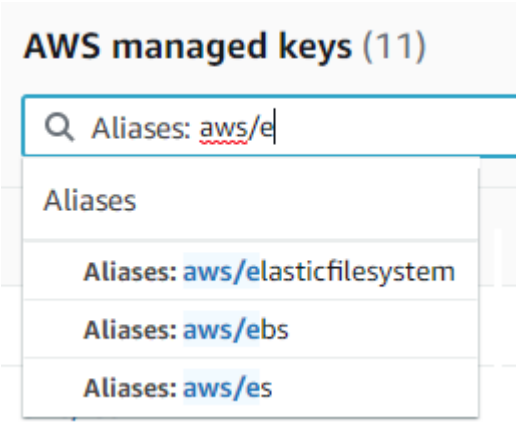
Vous pouvez filtrer les clés KMS en fonction de leurs valeurs de propriété ou de leurs balises. Le filtre s'applique à toutes les clés KMS de la table, même si elles n'apparaissent pas sur la page actuelle de la table. Le filtre n'est pas sensible à la casse.

Les propriétés pouvant être filtrées sont répertoriées dans la zone de filtre. Dans la page Clés gérées par AWS , vous pouvez filtrer par alias et ID de clé. Sur la page Clés gérées par le client, vous pouvez filtrer en fonction de leurs propriétés Alias, ID de clé et Type de clé et en fonction de leurs balises.

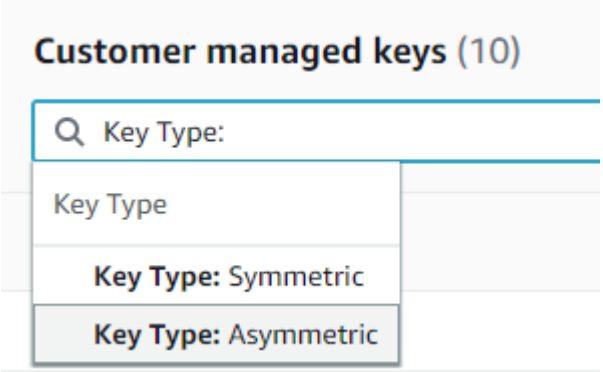
- Dans la page Clés gérées par AWS , vous pouvez filtrer par alias et ID de clé.
- Sur la page Clés gérées par le client, vous pouvez filtrer en fonction des balises ou de leurs propriétés Alias, ID de clé et Type de clé et de régionalité.

Pour filtrer en fonction d'une valeur de propriété, choisissez le filtre, le nom de la propriété, puis une valeur dans la liste des valeurs de propriété réelles. Pour filtrer par balise, choisissez la clé de balise, puis choisissez dans la liste des valeurs réelles de balise. Après avoir choisi une clé de propriété ou une clé de balise, vous pouvez également saisir l'ensemble ou une partie de la valeur de propriété ou de la balise. Vous verrez un aperçu des résultats avant de faire votre choix.

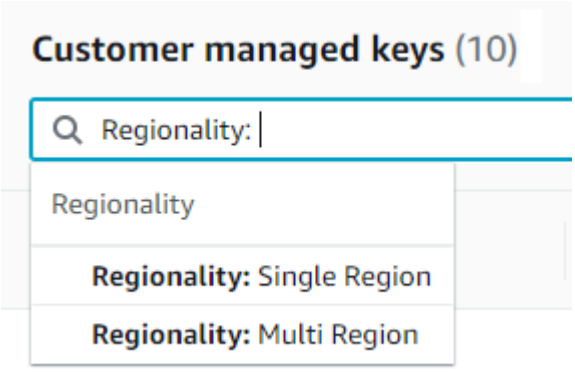
Par exemple, pour afficher les clés KMS avec un nom d'alias qui contient `aws/e`, choisissez la zone de filtre, choisissez Alias, saisissez `aws/e`, puis appuyez sur `Enter` ou `Return` pour ajouter le filtre.



Pour afficher uniquement les clés KMS asymétriques sur la page Clés gérées par le client, cliquez sur la zone de filtre, choisissez Key type (Type de clé) puis Key type: Asymmetric (Type de clé : Asymétrique). L'option Asymmetric (Asymétrique) apparaît uniquement lorsque vous avez des clés KMS asymétriques dans le tableau. Pour en savoir plus sur l'identification des clés KMS asymétriques, veuillez consulter Identification des clés KMS asymétriques.



Pour afficher uniquement les touches multi-région, dans la page Clés gérées par le client, choisissez la zone de filtre, puis Regionality (Régionalité) et Regionality: Multi-Region (Régionalité : Multi-régions). L'option Multi-Region (Multi-régions) apparaît uniquement lorsque vous avez des clés multi-région dans la table. Pour en savoir plus sur l'identification des clés multi-région, veuillez consulter Affichage des clés multi-régions.

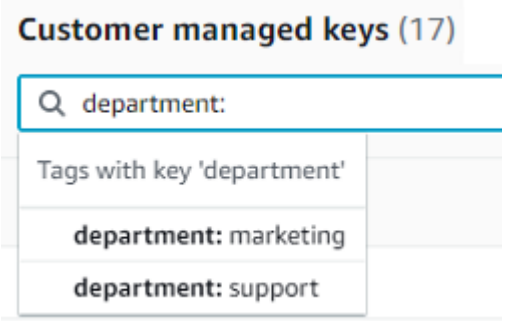


Le filtrage des balises est un peu différent. Pour afficher uniquement les clés KMS avec une balise particulière, choisissez la zone de filtre, choisissez la clé de balise, puis choisissez parmi les valeurs réelles de balise. Vous pouvez également saisir l'ensemble ou une partie de la valeur de balise.

Le tableau résultant affiche toutes les clés KMS avec la balise choisie. Cependant, il n'affiche pas la balise. Pour afficher la balise, choisissez l'ID de clé ou l'alias de la clé KMS et, sur sa page de détails, choisissez l'option Tags (Balises). Les onglets apparaissent sous la section General configuration (Configuration générale).

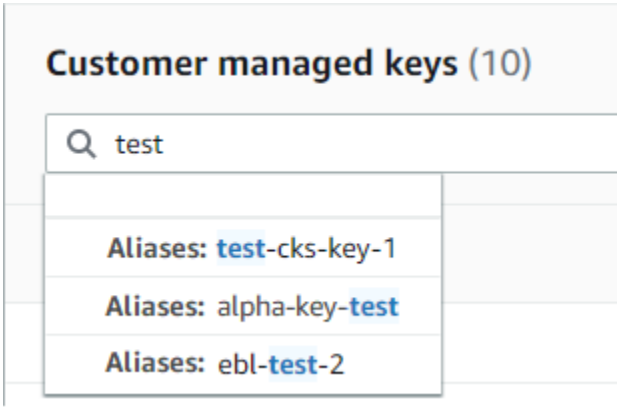
Ce filtre nécessite à la fois la clé de balise et la valeur de balise. Il ne trouvera pas de clés KMS en tapant uniquement la clé de balise ou seulement sa valeur. Pour filtrer les balises en fonction de la totalité ou d'une partie de la clé ou de la valeur de balise, utilisez l'ListResourceTagsopération pour obtenir les clés KMS balisées, puis utilisez les fonctionnalités

de filtrage de votre langage de programmation. Pour obtenir un exemple, consultez ListResourceTags: Récupère les tags des clés KMS.

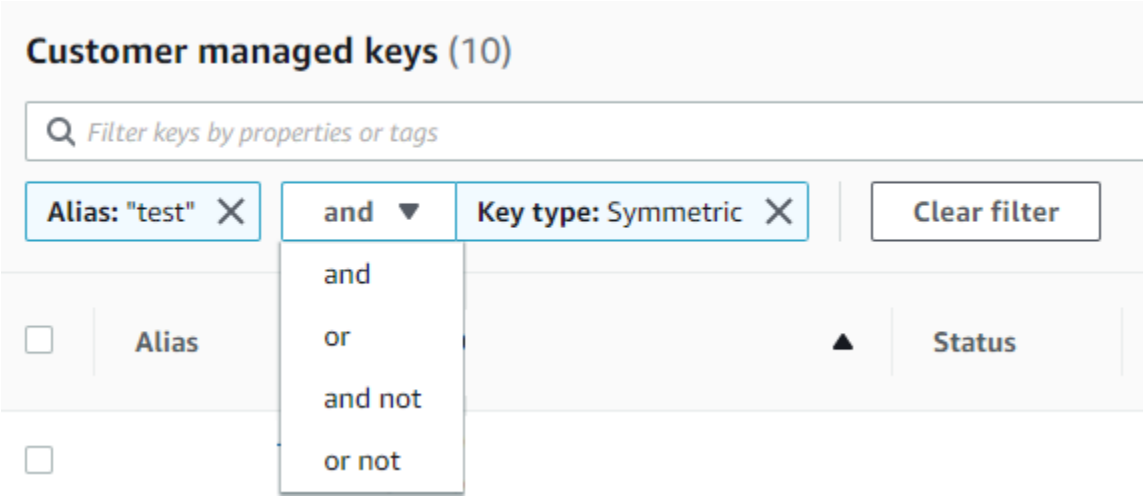


Pour rechercher du texte, dans la zone de filtre, saisissez l'ensemble ou une partie d'alias, d'ID de clé, d'un type de clé ou d'une clé de balise. (Après avoir sélectionné la clé de balise, vous pouvez rechercher une valeur de balise). Vous verrez un aperçu des résultats avant de faire votre choix.

Par exemple, pour afficher les clés KMS avec `test` dans ses clés de balise ou ses propriétés filtrables, saisissez `test` dans la zone de filtre. La prévisualisation affiche les clés KMS que le filtre sélectionnera. Dans ce cas, `test` apparaît uniquement dans la propriété Alias.



Vous pouvez utiliser plusieurs filtres en même temps. Lorsque vous ajoutez des filtres supplémentaires, vous pouvez également sélectionner un opérateur logique.

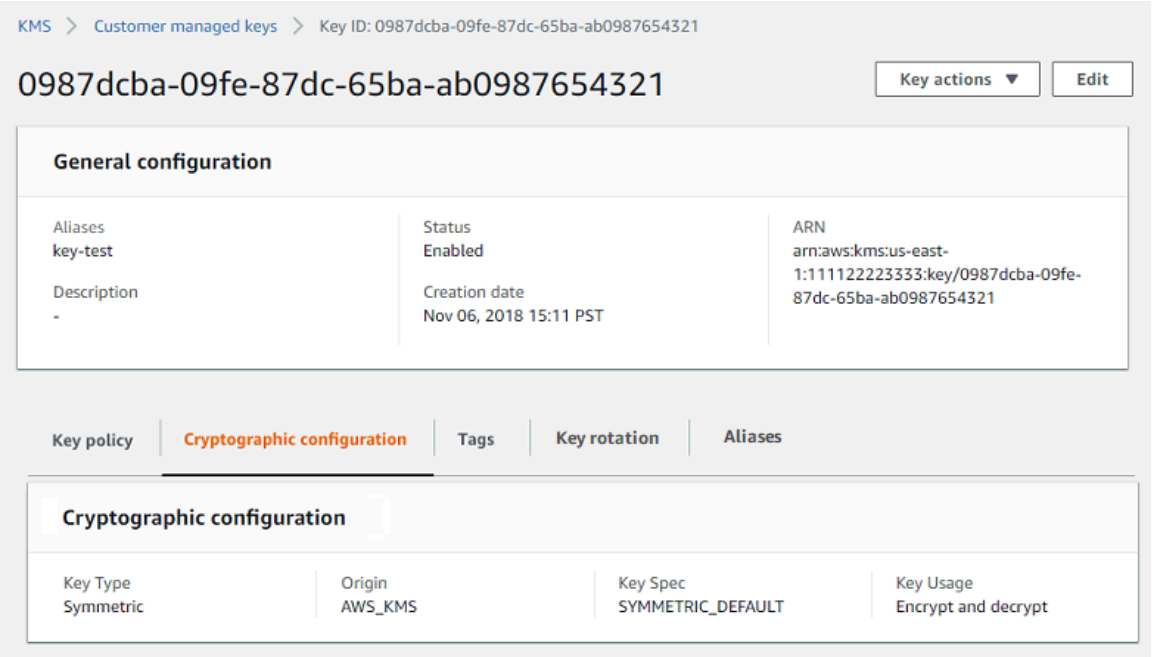


Affichage des détails de clé KMS

La page des détails de chaque clé KMS affiche les propriétés de la clé KMS. Elle diffère légèrement selon les différents types de clés KMS.

Pour afficher des informations détaillées sur une clé KMS, sur la page Clés gérées par AWS ou Clés gérées par le client, choisissez l'alias ou l'ID de clé de la clé KMS.

La page de détails d'une clé KMS inclut une section General Configuration (Configuration générale) qui affiche les propriétés de base de la clé KMS. Elle contient également des onglets sur lesquels vous pouvez afficher et modifier les propriétés de la clé KMS, telles que sa politique de clé, sa configuration de chiffrement, les balises, les éléments de clé (pour les clés KMS avec les éléments de clé importés), la rotation de clé (pour les clés KMS de chiffrement symétriques), le type de région (pour les clés multi-région) et sa clé publique (pour les clés KMS asymétriques).



La liste suivante décrit les champs de l'affichage détaillé, y compris les champs dans les onglets. Certains de ces champs sont également disponibles sous forme de colonnes dans l'affichage de la table.

Alias

Où : Onglet Alias

Un nom convivial pour la clé KMS. Vous pouvez utiliser un alias pour identifier la clé KMS dans la console et dans certaines API AWS KMS. Pour plus de détails, consultez Utilisation des alias.

L'onglet Alias affiche tous les alias associés à la clé KMS dans le Compte AWS et la région.

ARN

Où : section General configuration (Configuration générale)

Amazon Resource Name (ARN) de la clé KMS. Cette valeur identifie de manière unique la clé KMS. Vous pouvez l'utiliser pour identifier la clé KMS dans les opérations d'API AWS KMS.

État de connexion

Indique si un magasin de clés personnalisé est connecté à son magasin de clés de sauvegarde. Ce champ ne s'affiche que lorsque la clé KMS est créée dans un magasin de clés personnalisé.

Pour plus d'informations sur les valeurs de ce champ, consultez ConnectionStatela référence de l'AWS KMSAPI.

Date de création

Où : section General configuration (Configuration générale)

Date et heure de création de la clé KMS. Cette valeur est affichée dans l'heure locale du périphérique. Le fuseau horaire ne dépend pas de la région.

Contrairement à Expiration, la création se réfère uniquement à la clé KMS, pas à ses éléments de clé.

ID de cluster CloudHSM

Où : onglet Cryptographic configuration (Configuration de chiffrement)

L'ID du cluster AWS CloudHSM qui contient les éléments de clé pour la clé KMS. Ce champ ne s'affiche que lorsque la clé KMS est créée dans un magasin de clés personnalisé.

Si vous choisissez l'ID de cluster CloudHSM, il ouvre la page Clusters dans la console AWS CloudHSM.

ID du magasin de clés personnalisé

Où : onglet Cryptographic configuration (Configuration de chiffrement)

L'ID du magasin de clés personnalisé qui contient la clé KMS. Ce champ ne s'affiche que lorsque la clé KMS est créée dans un magasin de clés personnalisé.

Si vous choisissez l'ID du magasin de clés personnalisé, il ouvre la page Magasins de clés personnalisés dans la console AWS KMS.

Nom du magasin de clés personnalisé

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Le nom du magasin de clés personnalisé qui contient la clé KMS. Ce champ ne s'affiche que lorsque la clé KMS est créée dans un magasin de clés personnalisé.

Type de magasin de clés personnalisé

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Indique si le magasin de clés personnalisé est un magasin de clés AWS CloudHSM ou un magasin de clés externe. Ce champ ne s'affiche que lorsque la clé KMS est créée dans un magasin de clés personnalisé.

Description

Où : section General configuration (Configuration générale)

Une brève description facultative de la clé KMS que vous pouvez écrire et modifier. Pour ajouter ou mettre à jour la description d'une clé gérée par le client, au-dessus de General Configuration (Configuration générale), choisissez Edit (Modifier).

Algorithmes de chiffrement

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Répertorie les algorithmes de chiffrement qui peuvent être utilisés avec la clé KMS dans AWS KMS. Ce champ s'affiche uniquement lorsque le Type de clé est Asymmetric (Asymétrique) et que Key usage (Utilisation de la clé) est Encrypt and decrypt (Chiffrer et déchiffrer). Pour de plus amples informations sur les algorithmes de chiffrement pris en charge par AWS KMS, veuillez consulter Spécification de clé SYMMETRIC_DEFAULT et Spécifications des clés RSA pour le chiffrement et le déchiffrement.

Date d’expiration

Où : onglet Key material (Éléments de clé)

La date et l'heure auxquelles les éléments de clé KMS expirent. Ce champ s'affiche uniquement pour les clés KMS avec des éléments de clé importés, c'est-à-dire lorsque l'Origine est Externe et que la clé KMS a des éléments de clé qui expirent.

ID de clé externe

Où : onglet Cryptographic configuration (Configuration de chiffrement)

L'ID de la clé externe associée à une clé KMS dans un magasin de clés externe. Ce champ ne s'affiche que pour les clés KMS dans un magasin de clés externe.

État de la clé externe

Où : onglet Cryptographic configuration (Configuration de chiffrement)

État le plus récent signalé par le proxy de magasin de clés externe pour la clé externe associée à la clé KMS. Ce champ ne s'affiche que pour les clés KMS dans un magasin de clés externe.

Utilisation d'une clé externe

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Opérations cryptographiques activées sur la clé externe associée à la clé KMS. Ce champ ne s'affiche que pour les clés KMS dans un magasin de clés externe.

Stratégie de clé

Où : Onglet Key policy (Politique de clé)

Contrôle l'accès à la clé KMS ainsi qu'aux politiques IAM et aux octrois. Chaque clé KMS a une politique de clé. C'est le seul élément d'autorisation obligatoire. Pour modifier la politique d'une clé KMS gérée par un client, sous l'onglet Key policy (Politique de clé), choisissez Edit (Modifier). Pour plus de détails, consultez the section called “Politiques de clé”.

Rotation des clés d'accès

Où : Onglet Key rotation (Rotation de clé)

Activer ou désactive la rotation automatique des éléments de clé dans une clé KMS gérée par le client. Pour modifier l'état de rotation d'une clé gérée par le client, cochez la case de l'onglet Rotation de clé.

Vous ne pouvez pas activer ou désactiver la rotation des éléments de clé dans une Clé gérée par AWS. Les Clés gérées par AWS sont automatiquement soumises à la rotation chaque année.

Spécifications de la clé

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Type d'élément de clé dans la clé KMS. AWS KMS prend en charge les clés KMS de chiffrement symétriques (SYMMETRIC_DEFAULT), les clés KMS HMAC de différentes longueurs, les clés KMS pour les clés RSA de différentes longueurs et les clés de courbe elliptique avec différentes courbes. Pour plus de détails, consultez Spécifications de la clé.

Type de clé

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Indique si la clé KMS est Symmetric (Symétrique) ou Asymmetric (Asymétrique).

Utilisation de la clé

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Indique si une clé KMS peut être utilisée pour Encrypt and decrypt (Chiffrer et déchiffrer), Sign and verify (Signer et vérifier) ou Generate and verify MAC (Générer et vérifier le MAC). Pour plus de détails, consultez Utilisation de la clé.

Origin

Où : onglet Cryptographic configuration (Configuration de chiffrement)

La source de l'élément de clé pour la clé KMS. Les valeurs valides sont :

- AWS KMS pour les éléments de clé que AWS KMS génère
- AWS CloudHSM pour les clés KMS dans le magasin de clés AWS CloudHSM
- External (Externe) pour les éléments de clé importés (BYOK)
- External key store (Magasin de clés externe) pour les clés KMS dans un magasin de clés externe

Algorithmes MAC

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Répertorie les algorithmes MAC qui peuvent être utilisés avec la clé KMS HMAC dans AWS KMS. Ce champ apparaît uniquement lorsque la spécification de clé est une spécification de clé HMAC (HMAC_*). Pour de plus amples informations sur les algorithmes MAC pris en charge par AWS KMS, veuillez consulter Spécifications de clé pour les clés KMS HMAC.

Clé primaire

Où : Onglet Regionality (Régionalité)

Indique que cette clé KMS est une clé primaire multi-région. Les utilisateurs autorisés peuvent utiliser cette section pour modifier la clé primaire en une autre clé multi-région associée. Ce champ apparaît uniquement lorsque la clé KMS est une clé principale multi-région.

Clé publique

Où : Onglet Public key (Clé publique)

Affiche la clé publique d'une clé KMS asymétrique. Les utilisateurs autorisés peuvent utiliser cet onglet pour copier et télécharger la clé publique.

Régionalité

Où : section General configuration (Configuration générale) et onglet Regionality (Régionalité)

Indique si une clé KMS est une clé de région unique, une clé primaire multi-région ou une clé de réplica multi-région. Ce champ apparaît uniquement lorsque la clé KMS est une clé multi-région.

Touches multi-région associées

Où : Onglet Regionality (Régionalité)

Affiche tous les clés primaires et de réplica multi-région, à l'exception de la clé KMS actuelle. Ce champ apparaît uniquement lorsque la clé KMS est une clé multi-région.

Dans la section Related multi-Region keys (Clés multi-région associées) d'une clé primaire, les utilisateurs autorisés peuvent créer des clés de réplica.

Clé de réplica

Où : Onglet Regionality (Régionalité)

Indique que cette clé KMS est une clé de réplica multi-région. Ce champ apparaît uniquement lorsque la clé KMS est une clé de réplica multi-région.

Algorithmes de signature

Où : onglet Cryptographic configuration (Configuration de chiffrement)

Répertorie les algorithmes de signature qui peuvent être utilisés avec la clé KMS dans AWS KMS. Ce champ s'affiche uniquement lorsque le Type de clé est Asymmetric (Asymétrique) et que Key usage (Utilisation de la clé) est Sign and verify (Signer et vérifier). Pour de plus amples informations sur les algorithmes de signature pris en charge par AWS KMS, veuillez consulter Spécifications des clés RSA pour la signature et la vérification et Spécifications de la clé de courbe elliptique.

Statut

Où : section General configuration (Configuration générale)

État de clé de la clé KMS. Vous pouvez utiliser la clé KMS dans les opérations de chiffrement uniquement lorsque l'état est Enabled (Activé). Pour obtenir une description détaillée de chaque état de clé KMS et de son effet sur les opérations que vous pouvez exécuter sur la clé KMS, veuillez consulter Principaux états des clés AWS KMS.

Balises

Où : Onglet Tags (Balises)

Paires clé-valeur facultatives décrivant la clé KMS. Pour ajouter ou modifier les balises d'une clé KMS, sous l'onglet Tags (Balises), choisissez Edit (Modifier).

Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour plus d'informations sur le balisage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.

Personnalisation de vos tables de clés KMS

Vous pouvez personnaliser les tables qui apparaissent dans les pages des Clés gérées par AWS et des clés gérées par le client dans la AWS Management Console pour répondre à vos besoins. Vous pouvez choisir les colonnes de la table, le nombre de AWS KMS keys sur chaque page (Taille de la page) et le retour à la ligne. La configuration que vous choisissez est enregistrée lorsque vous la confirmez et réappliquée chaque fois que vous ouvrez les pages.

Personnalisation de vos tables de clés KMS

1. Sur la page des Clés gérées par AWS ou des clés gérées par le client, choisissez l'icône des paramètres

   ( )

   dans le coin supérieur droit de la page.
2. Sur la page Préférences, choisissez vos paramètres préférés, puis choisissez Confirmer.

Pensez à utiliser le paramètre Page size (Taille de page) pour augmenter le nombre de clés KMS affichées sur chaque page, surtout si vous utilisez généralement un périphérique facile à faire défiler.

Les colonnes de données que vous affichez peuvent varier en fonction de la table, de votre rôle de tâche et des types de clés KMS dans le compte et la région. La table suivante propose quelques configurations. Pour obtenir une description des colonnes, veuillez consulter Affichage des détails de clé KMS.

Configurations de tables de clés KMS suggérées

Vous pouvez personnaliser les colonnes qui apparaissent dans votre table de clés KMS pour afficher les informations dont vous avez besoin sur vos clés KMS.

Clés gérées par AWS

Par défaut, la table des Clé gérée par AWS affiche les colonnes Alias, Key ID (ID de clé) et Status (État). Ces colonnes sont idéales pour la plupart des cas d'utilisation.

Clés KMS de chiffrement symétriques

Si vous utilisez uniquement des clés KMS de chiffrement symétriques avec des éléments de clé générés par AWS KMS, les colonnes Alias, Key ID (ID de clé), Status (État) et Creation date (Date de création) sont susceptibles d'être les plus utiles.

Clés KMS asymétriques

Si vous utilisez des clés KMS asymétriques, en plus des colonnes Alias, Key ID (ID de clé) et Status (État), envisagez d'ajouter les colonnes Key type (Type de clé), Key spec (Spécifications de la clé) et Key usage (Utilisation de la clé). Ces colonnes indiquent si une clé KMS est symétrique ou asymétrique, le type d'élément de clé et si la clé KMS peut être utilisée pour le chiffrement ou la signature.

Clés KMS HMAC

Si vous utilisez des clés KMS HMAC, en plus des colonnes Alias, Key ID (ID de clé) et Status (État), envisagez d'ajouter les colonnes Key spec (Spécifications de la clé) et Key usage (Utilisation de la clé). Ces colonnes vous indiqueront si une clé KMS est une clé HMAC. Étant donné que vous ne pouvez pas trier les clés KMS en fonction des spécifications de la clé ou de l'utilisation de la clé, utilisez des alias et des balises pour identifier vos clés HMAC, puis utilisez les fonctions de filtre de la console AWS KMS pour filtrer par alias ou par balises.

Éléments de clé importés

Si vous avez des clés KMS avec des éléments de clé importés, envisagez d'ajouter les colonnes Origin (Origine) et Expiration date (Date d'expiration). Ces colonnes vous indiqueront si l'élément de clé dans une clé KMS est importé ou généré par AWS KMS et quand les éléments de clé expirent, le cas échéant. Le champ Creation date (Date de création) affiche la date à laquelle la clé KMS a été créée (sans élément de clé). Il ne reflète aucune caractéristique de l'élément de clé.

Clés dans des magasins de clés personnalisés

Si vous avez des clés KMS dans des magasins de clés personnalisés, envisagez d'ajouter les colonnes Origin (Origine) et Custom key store ID (ID du magasin de clés personnalisé). Ces colonnes indiquent que la clé KMS se trouve dans un magasin de clés personnalisé, identifient celui-ci et affichent son type.

Clés multi-région

Si vous disposez de clés multi-région, envisagez d'ajouter la colonne Regionality (Régionalité). Cela indique si une clé KMS est une clé de région unique, une clé primaire multi-région ou une clé de réplica multi-région.

Affichage des clés KMS avec l'API

Vous pouvez utiliser l'API AWS Key Management Service (AWS KMS) pour afficher vos clés CMK. Cette section présente plusieurs opérations qui renvoient des informations sur les clés KMS existantes. Les exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Rubriques

- ListKeys: Obtenez l'ID et l'ARN de toutes les clés KMS
- DescribeKey: obtenir des informations détaillées sur une clé KMS
- GetKeyPolicy: Obtenez la politique de clé attachée à une clé KMS
- ListAliases: Obtenir les noms d'alias et les ARN pour les clés KMS
- ListResourceTags: Récupère les tags des clés KMS

ListKeys: Obtenez l'ID et l'ARN de toutes les clés KMS

L'ListKeysopération renvoie l'ID et le nom de ressource Amazon (ARN) de toutes les clés KMS du compte et de la région.

Par exemple, cet appel à l'opération `ListKeys` renvoie l'ID et l'ARN de chaque clé KMS de ce compte fictif. Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Obtention des ID de clé et des ARN de clé des clés KMS.

```
$ aws kms list-keys

{
  "Keys": [
    {
      "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
      "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
    },
    {
```

```
      "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
      "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
}
```

DescribeKey: obtenir des informations détaillées sur une clé KMS

L'DescribeKeyopération renvoie des informations sur la clé KMS spécifiée. Pour identifier la clé KMS, utilisez son ID de clé, son ARN de clé, son nom d'alias ou son ARN d'alias.

Contrairement à l'ListKeysopération, qui affiche uniquement les clés KMS dans le compte et la région de l'appelant, les utilisateurs autorisés peuvent utiliser l'`DescribeKey`opération pour obtenir des informations sur les clés KMS d'autres comptes.

> **Note**
>
> La réponse `DescribeKey` inclut à la fois les membres `KeySpec` et `CustomerMasterKeySpec` avec les mêmes valeurs. Le membre `CustomerMasterKeySpec` est obsolète.

Par exemple, cet appel à `DescribeKey` renvoie des informations sur une clé KMS de chiffrement symétrique. Les champs de la réponse varient en fonction des spécifications AWS KMS key, de l'état de la clé et de l'origine des éléments de clé. Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Affichage d'un AWS KMS key.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1499988169.234,
```

```
        "MultiRegion": false,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Cet exemple appelle une opération `DescribeKey` sur une clé KMS asymétrique utilisée pour la signature et la vérification. La réponse inclut les algorithmes de signature que AWS KMS prend en charge pour cette clé KMS.

```
$ aws kms describe-key --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
    "KeyMetadata": {
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "Origin": "AWS_KMS",
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "CreationDate": 1569973196.214,
        "Description": "",
        "KeySpec": "ECC_NIST_P521",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "AWSAccountId": "111122223333",
        "Enabled": true,
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

GetKeyPolicy: Obtenez la politique de clé attachée à une clé KMS

L'GetKeyPolicyopération obtient la politique de clé attachée à la clé KMS. Pour identifier la clé KMS, utilisez son ID de clé ou son ARN de clé. Vous devez également spécifier le nom de la politique, qui

est toujours par défaut `default`. (Si la sortie est difficile à lire, ajoutez l'option `--output text` à votre commande.) `GetKeyPolicy` fonctionne uniquement sur les clés KMS du compte et de la région de l'appelant.

Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Obtention d'une politique de clé.

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default

{
  "Version" : "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [ {
    "Sid" : "Enable IAM User Permissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

ListAliases: Obtenir les noms d'alias et les ARN pour les clés KMS

L'ListAliasesopération renvoie les alias du compte et de la région. Le `TargetKeyId` de la réponse indique l'ID de clé de la clé KMS auquel l'alias fait référence, le cas échéant.

Par défaut, la commande ListAliases retourne tous les alias figurant dans le compte et la région. Cela inclut les alias que vous avez créés et associés à vos clés gérées par le client, et les alias créés par AWS et associés à votre Clé gérée par AWS dans votre compte. Vous pouvez reconnaître les alias AWS car leurs noms ont le format `aws/<service-name>`, par exemple `aws/dynamodb`.

La réponse peut également inclure les alias ne disposant pas de champ `TargetKeyId`, comme l'alias `aws/redshift` dans cet exemple. Ces alias prédéfinis ont été créés par AWS qui ne les a pas encore associés à une clé KMS.

Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Établissement de la liste des alias.

```
$ aws kms list-aliases
```

```
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasName": "alias/financeKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
        },
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
```

```
        },
        {
            "AliasName": "alias/aws/redshift",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/redshift"
        },
    ]
}
```

Pour obtenir les alias qui font référence à une clé KMS spécifique, utilisez le paramètre `KeyId`. La valeur du paramètre peut être l'ID de clé ou l'ARN de clé. Vous ne pouvez pas spécifier de nom d'alias ou d'ARN d'alias.

La commande de l'exemple suivant obtient les alias qui font référence à une clé KMS gérée par le client. Toutefois, vous pouvez utiliser une commande similaire pour rechercher les alias faisant également référence à des Clés gérées par AWS.

```
$ aws kms list-aliases --key-id arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "AliasName": "alias/financeKey",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
    ]
}
```

Pour obtenir uniquement les alias pour les Clés gérées par AWS, utilisez les fonctions de votre langage de programmation pour filtrer la réponse.

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

ListResourceTags: Récupère les tags des clés KMS

L'ListResourceTagsopération renvoie les balises de la clé KMS spécifiée. L'API renvoie des balises pour une clé KMS, mais vous pouvez exécuter la commande dans une boucle pour obtenir des balises pour toutes les clés KMS du compte et de la région, ou pour un ensemble de clés KMS que vous sélectionnez. Cette API renvoie une page à la fois, donc si vous avez de nombreuses balises sur de nombreuses clés KMS, vous devrez peut-être utiliser le paginateur dans votre langage de programmation pour obtenir toutes les balises que vous souhaitez.

L'opération `ListResourceTags` renvoie des balises pour toutes les clés KMS, mais les Clé gérée par AWS ne sont pas labelisées. Cela fonctionne uniquement sur les clés KMS du compte et de la région de l'appelant.

Pour trouver les balises d'une clé KMS, utilisez l'opération `ListResourceTags`. Le paramètre `KeyId` est obligatoire. Il accepte un ID de clé ou un ARN de clé. Avant d'exécuter cet exemple, remplacez l'ARN de clé de l'exemple par un ARN valide.

```
$  aws kms list-resource-tags --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
      {
    "Tags": [
        {
            "TagKey": "Department",
            "TagValue": "IT"
        },
        {
            "TagKey": "Purpose",
            "TagValue": "Test"
        }
    ],
    "Truncated": false
}
```

Vous pouvez également, si vous le souhaitez, utiliser l'opération `ListResourceTags` pour obtenir toutes les clés KMS du compte et de la région avec une balise, une clé de balise ou une valeur de balise particulière. Pour ce faire, utilisez les fonctions de filtrage de votre langage de programmation.

Par exemple, le script Bash suivant utilise les `ListResourceTags` opérations ListKeyset pour obtenir toutes les clés KMS du compte et de la région avec une clé de `Project` balise. Ces deux opérations n'obtiennent que la première page de résultats. Si vous disposez de nombreuses clés

KMS ou de nombreuses balises, utilisez les fonctions de pagination de votre langue pour obtenir le résultat complet de chaque opération. Avant d'exécuter cet exemple, remplacez l'ID de clé de l'exemple par un ID valide.

```
TARGET_TAG_KEY='Project'

for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text); do
  key_tags=$(aws kms list-resource-tags --key-id "$key" --query "Tags[?TagKey==\`
$TARGET_TAG_KEY\`]")
  if [ "$key_tags" != "[]" ]; then
    echo "Key: $key"
    echo "$key_tags"
  fi
done
```

La sortie est formatée comme l'exemple de sortie suivant.

```
Key: 0987dcba-09fe-87dc-65ba-ab0987654321
[
    {
        "TagKey": "Project",
        "TagValue": "Gamma"
    }
]
Key: 1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
[
    {
        "TagKey": "Project",
        "TagValue": "Alpha"
    }
]
Key: 0987ab65-43cd-21ef-09ab-87654321cdef
[
    {
        "TagKey": "Project",
        "TagValue": "Alpha"
    }
]
```

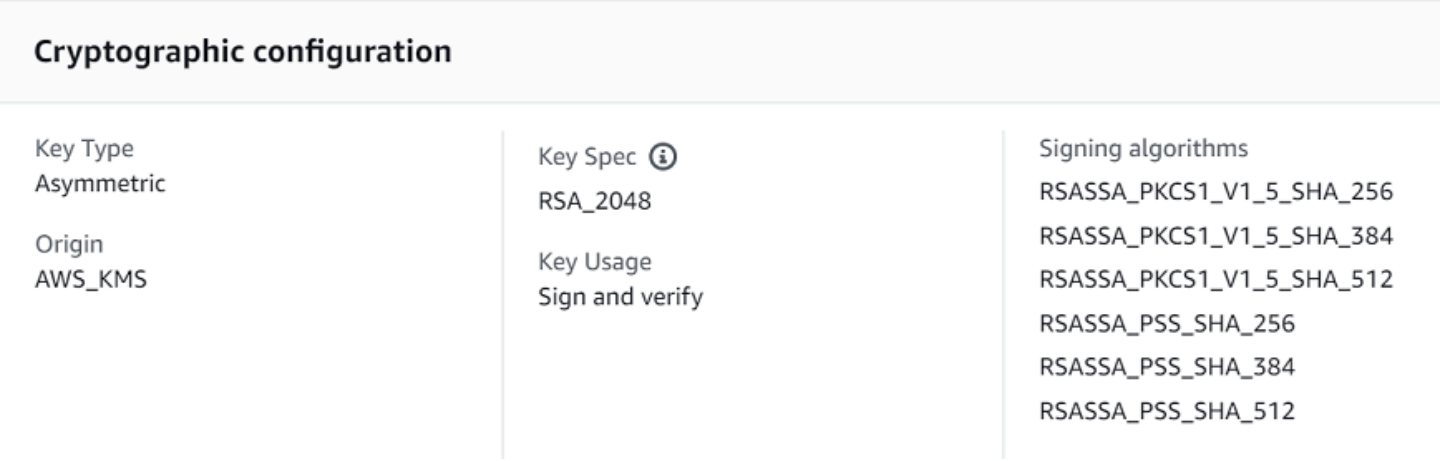
Affichage de la configuration de chiffrement des clés KMS

Après avoir créé votre clé KMS, vous pouvez afficher sa configuration de chiffrement. Vous ne pouvez pas modifier la configuration d'une clé KMS après sa création. Si vous préférez une configuration différente, supprimez la clé KMS et créez-la à nouveau.

Vous pouvez trouver la configuration de chiffrement de vos clés KMS, inclure les spécifications de la clé, l'utilisation de la clé et les algorithmes de chiffrement ou de signature pris en charge dans la console AWS KMS ou à l'aide de l'API AWS KMS. Pour plus de détails, consultez Identification des clés KMS asymétriques.

Dans la console AWS KMS, la page de détails de chaque clé KMS comprend un onglet Cryptographic configuration (Configuration de chiffrement) qui affiche des détails de chiffrement sur vos clés KMS. Par exemple, l'image suivante montre l'onglet Cryptographic configuration (Configuration de chiffrement) d'une clé KMS RSA utilisée pour la signature et la vérification.

L'onglet Cryptographic configuration (Configuration cryptographique) de certaines clés KMS à usage spécial comporte des sections spécialisées supplémentaires. Par exemple, l'onglet Cryptographic configuration (Configuration cryptographique) d'une clé KMS dans un magasin de clés personnalisé comporte une section Custom key stores (Magasins de clés personnalisés). L'onglet Cryptographic configuration (Configuration cryptographique) d'une clé KMS dans un magasin de clés externe comporte une section External key (Clé externe).



Dans l'AWS KMSAPI, utilisez l'DescribeKeyopération. La structure `KeyMetadata` de la réponse comprend la configuration de chiffrement de la clé KMS. Par exemple, `DescribeKey` renvoie la réponse suivante pour une clé KMS RSA utilisée pour la signature et la vérification.

```
{
```

```
  "KeyMetadata": {

    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": 1571767572.317,
    "CustomerMasterKeySpec": "RSA_2048",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "KeySpec": "RSA_2048",
    "KeyUsage": "SIGN_VERIFY",
    "SigningAlgorithms": [
        "RSASSA_PKCS1_V1_5_SHA_256",
        "RSASSA_PKCS1_V1_5_SHA_384",
        "RSASSA_PKCS1_V1_5_SHA_512",
        "RSASSA_PSS_SHA_256",
        "RSASSA_PSS_SHA_384",
        "RSASSA_PSS_SHA_512"
    ]
  }
}
```

Recherche de l'ID et de l'ARN d'une clé

Pour identifier une AWS KMS key, vous pouvez utiliser l'ID de clé ou le nom Amazon Resource Name (ARN de clé). Dans les opérations de chiffrement, vous pouvez également utiliser le nom d'alias ou l'ARN d'alias.

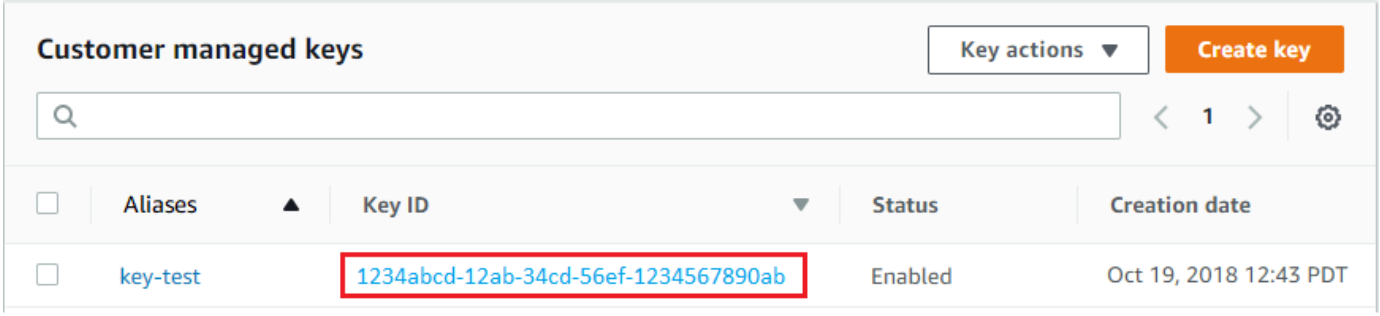
Pour de plus amples informations sur les identificateurs de clé KMS pris en charge par AWS KMS, veuillez consulter Identifiants clés () KeyId. Pour obtenir de l'aide afin de trouver un nom d'alias et un ARN d'alias, veuillez consulter Recherche du nom d'alias et de l'ARN d'alias.

Pour trouver l'ID et l'ARN de la clé (console)

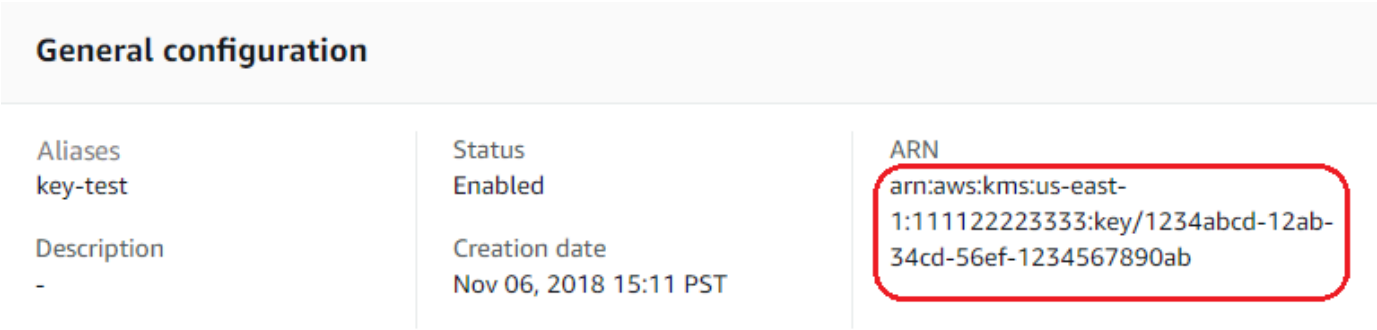
1. Ouvrez la console AWS KMS à l’adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de Région dans l’angle supérieur droit de la page.

3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client. Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS.
4. Pour obtenir l'ID de clé d'une clé KMS, veuillez consulter la ligne qui commence par l'alias de clé KMS.

   Par défaut, la colonne ID de clé apparaît dans les tables. Si la colonne ID de clé n'apparaît pas dans votre table, utilisez la procédure décrite à la section the section called “Personnalisation de vos tables de clés KMS” pour la restaurer. Vous pouvez également afficher l'ID de clé d'une clé KMS sur sa page de détails.



5. Pour rechercher l'Amazon Resource Name (ARN) de la clé KMS, choisissez l'ID de clé ou l'alias. L'ARN de clé apparaît dans la section Configuration générale.



Pour rechercher l'ID de clé et l'ARN de clé (API AWS KMS)

Pour trouver l'ID de clé et l'ARN de clé d'unAWS KMS key, utilisez l'ListKeysopération. Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Obtention des ID de clé et des ARN et Obtenir les ID de clé et les ARN de clé.

La réponse `ListKeys` inclut l'ID de clé et l'ARN de clé pour chaque clé KMS du compte et de la région.

```
$ aws kms list-keys
{
    "Keys": [
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ]
}
```

Recherche du nom d'alias et de l'ARN d'alias

Un alias est un nom convivial pour une AWS KMS keys AWS KMS (clé KMS). Vous pouvez trouver le nom d'alias et l'ARN d'alias dans la console AWS KMS ou l'API AWS KMS.

Pour de plus amples informations sur les identificateurs de clé KMS pris en charge par AWS KMS, veuillez consulter Identifiants clés () KeyId. Pour obtenir de l'aide sur la recherche de l'ID de clé et l'ARN de clé, veuillez consulter Recherche de l'ID et de l'ARN d'une clé.

Rubriques

- Pour trouver le nom d'alias et l'ARN d'alias (console)
- Pour trouver le nom d'alias et l'ARN d'alias (API AWS KMS)

Pour trouver le nom d'alias et l'ARN d'alias (console)

La console AWS KMS affiche les alias associés à la clé KMS.

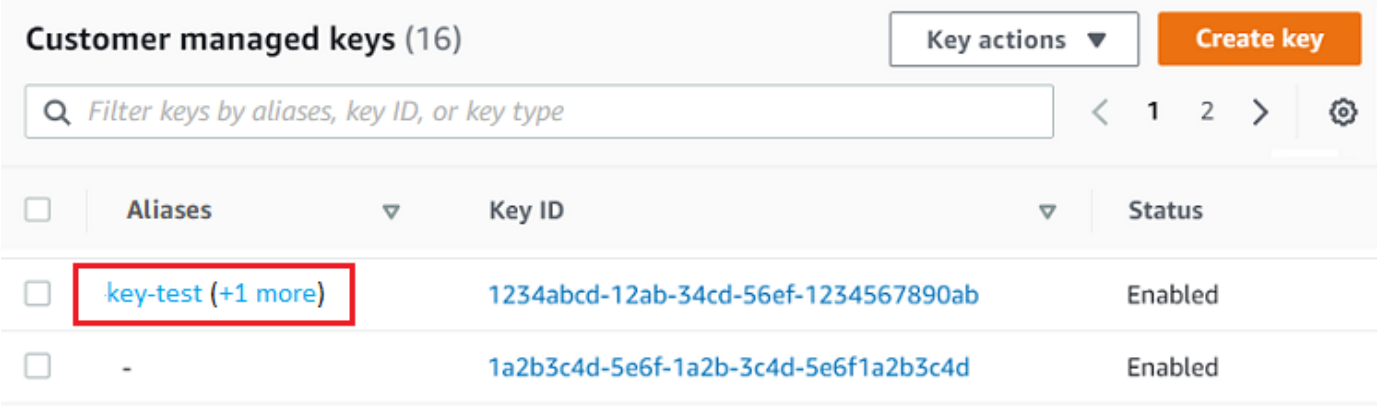
1. Ouvrez la console AWS KMS à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de Région dans l'angle supérieur droit de la page.
3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client. Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS.

4. La colonne Aliases (Alias) affiche l'alias de chaque clé KMS. Si une clé KMS n'a pas d'alias, un tiret (-) apparaît dans la colonne Aliases (Alias).

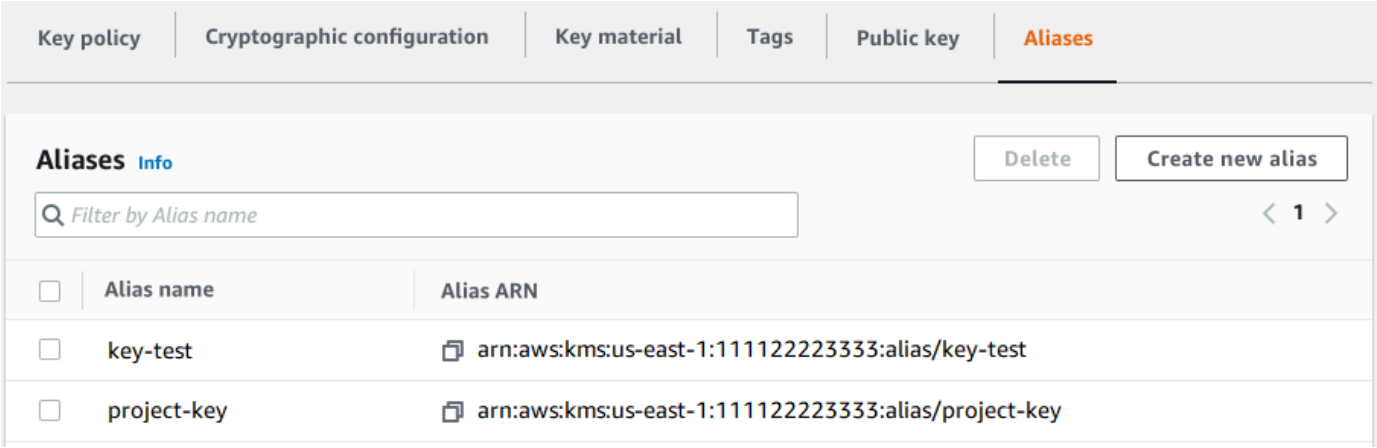
   Si une clé KMS possède plusieurs alias, la colonne Aliases (Alias) contient également un résumé d'alias, tel que (+n plus). Par exemple, la clé KMS suivante a deux alias, dont l'un est `key-test`.

   Pour trouver le nom d'alias et l'ARN d'alias de tous les alias de la clé KMS, utilisez l'onglet Aliases (Alias).

   - Pour accéder directement à l'onglet Aliases (Alias), dans la colonne Aliases (Alias), choisissez le résumé de l'alias (+n plus). Un résumé d'alias apparaît uniquement si la clé KMS comporte plusieurs alias.
   - Vous pouvez également choisir l'alias ou l'ID de clé de la clé KMS (qui ouvre la page des détails de la clé KMS), puis l'onglet Aliases (Alias). Les onglets se trouvent sous la section General configuration (Configuration générale).



5. L'onglet Aliases (Alias) affiche le nom d'alias et l'ARN d'alias de tous les alias d'une clé KMS. Vous pouvez également créer et supprimer des alias pour la clé KMS sur cet onglet.



Pour trouver le nom d'alias et l'ARN d'alias (API AWS KMS)

Pour trouver le nom d'alias et l'ARN d'alias d'unAWS KMS key, utilisez l'ListAliasesopération. Pour obtenir des exemples dans plusieurs langages de programmation, veuillez consulter Établissement de la liste des alias et Obtenir les noms d'alias et les ARN.

Par défaut, la réponse inclut le nom d'alias et l'ARN d'alias pour chaque alias du compte et de la région. Pour obtenir uniquement les alias d'une clé KMS spécifique, utilisez le paramètre `KeyId`.

Par exemple, la commande suivante obtient uniquement les alias d'un exemple de clé KMS avec l'ID de clé `1234abcd-12ab-34cd-56ef-1234567890ab`.

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/key-test",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/key-test",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1593622000.191,
            "LastUpdatedDate": 1593622000.191
        },
        {
            "AliasName": "alias/project-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        }
    ]
}
```

Modification des clés

Vous pouvez modifier les propriétés suivantes de vos clés gérées par le client dans la console AWS KMS et en utilisant l'API AWS KMS.

Vous ne pouvez modifier aucune des propriétés de Clés gérées par AWS ou de Clés détenues par AWS. Ces clés sont gérées par les services AWS qui les ont créés.

Description

Vous pouvez modifier la description de votre clé gérée par le client sur la page de détails de la clé KMS ou en utilisant l'UpdateKeyDescriptionopération.

Pour modifier la description de clé dans la console, dans le coin supérieur droit de la page des détails de la clé KMS, choisissez Edit (Modifier).

Stratégie de clé

Vous pouvez modifier la politique clé dans l'onglet Politique clé de la page de détails de la clé gérée par le client ou en utilisant l'PutKeyPolicyopération.

Pour plus de détails, consultez Modification d'une politique de clé.

Balises

Vous pouvez créer et supprimer des balises sur les clés gérées par le client sur la page de la AWS KMSconsole, ou sur l'onglet Balises de la page de détails pour la clé gérée par le client. Vous pouvez également utiliser les UntagResourceopérations TagResourceet.

Pour plus de détails, veuillez consulter Clés de balisage.

Activer et désactiver

Vous pouvez activer et désactiver les clés KMS sur la page de la consoleAWS KMS Clés gérées par le client ou sur la page de détails pour la clé gérée par le client. Vous pouvez également utiliser les DisableKeyopérations EnableKeyet.

Pour plus de détails, consultez Activation et désactivation des clés.

Rotation automatique des clés

Vous pouvez activer et désactiver la rotation automatique des clés dans l'onglet Rotation des clés de la page de détails de la clé gérée par le client ou en utilisant les DisableKeyRotationopérations EnableKeyRotationet.

Pour plus de détails, consultez Rotation des AWS KMS keys.

Consultez aussi

Mise à jour des alias

Clés de balisage

Dans AWS KMS, vous pouvez ajouter des balises à une clé gérée par le client lorsque vous créez la clé KMS de clé, et étiqueter ou désétiqueter les clés KMS existantes à moins qu'elles ne soient en cours de suppression. Vous ne pouvez pas étiqueter d'alias, de magasins de clés personnalisés, de Clés gérées par AWS, de Clés détenues par AWS, ou de clés KMS dans d'autres Comptes AWS. Les balises sont facultatives, mais peuvent être très utiles.

Pour plus d'informations, consultez Création de clés et Modification des clés. Pour obtenir des informations générales sur les balises, y compris les bonnes pratiques, les politiques d'étiquetage, ainsi que le format et la syntaxe des balises, veuillez consulter Étiquetage des ressources AWS dans Référence générale d'Amazon Web Services.

Rubriques

- À propos des balises dans AWS KMS
- Gestion des balises de clé KMS dans la console
- Gestion des balises de clés KMS avec les opérations API
- Contrôle de l'accès aux balises
- Utilisation de balises pour contrôler l'accès aux clés KMS

À propos des balises dans AWS KMS

Une balise est un label de métadonnée que vous attribuez (ou que AWS peut attribuer) à une ressource AWS. Chaque balise est constituée d'une clé de balise et d'un valeur de balise, qui sont toutes deux des chaînes sensibles à la casse. La valeur de balise peut être une chaîne vide (nulle). Chaque balise d'une ressource doit avoir une clé de balise différente, mais vous pouvez ajouter la même balise à plusieurs ressources AWS. Chaque ressource peut avoir jusqu'à 50 balises créées par l'utilisateur.

N'incluez pas d'informations confidentielles ou sensibles dans la clé de balise ou la valeur de balise. Les étiquettes accessibles à de nombreux Services AWS, y compris la facturation.

Dans AWS KMS, vous pouvez ajouter des balises à une clé gérée par le client lorsque vous créez la clé KMS de clé, et étiqueter ou désétiqueter les clés KMS existantes à moins qu'elles ne soient en cours de suppression. Vous ne pouvez pas étiqueter d'alias, de magasins de clés personnalisés, de Clés gérées par AWS, de Clés détenues par AWS, ou de clés KMS dans d'autres Comptes AWS. Les balises sont facultatives, mais peuvent être très utiles.

Par exemple, vous pouvez ajouter une balise `"Project"="Alpha"` à toutes les clés KMS et compartiments Amazon S3 que vous utilisez pour le projet Alpha.

```
TagKey   = "Project"
TagValue = "Alpha"
```

Pour obtenir des informations générales sur les balises, y compris le format et la syntaxe, veuillez consulter Étiquetage des ressources AWS dans Référence générale d'Amazon Web Services.

Les balises vous permettent d'effectuer les actions suivantes :

- Identifier et organiser vos ressources AWS. De nombreux services AWS prennent en charge le balisage. Vous pouvez donc attribuer la même balise à des ressources à partir de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à une clé KMS et à un volume Amazon Elastic Block Store (Amazon EBS) ou un secret AWS Secrets Manager. Vous pouvez également utiliser des balises pour identifier les clés KMS pour l'automatisation.
- Suivre vos coûts AWS. Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Vous pouvez utiliser cette fonction pour suivre les coûts AWS KMS d'un projet, d'une application ou d'un centre de coûts.

  Pour en savoir plus sur l'utilisation des balises pour la répartition des coûts, veuillez consulter Utilisation des balises de répartition des coûts dans le Guide de l'utilisateur AWS Billing. Pour obtenir des informations sur les règles des clés et valeurs de balise, veuillez consulter Restrictions encadrant les balises définies par l'utilisateur dans le Guide de l'utilisateur AWS Billing.
- Contrôler l'accès à vos ressources AWS. Autoriser et refuser l'accès aux clés KMS en fonction de leurs balises fait partie de la prise en charge de AWS KMS pour le contrôle d'accès basé sur les attributs (ABAC). Pour plus d'informations sur le contrôle d'accès aux AWS KMS keys basé sur des balises, veuillez consulter Utilisation de balises pour contrôler l'accès aux clés KMS. Pour plus d'informations générales sur l'utilisation des balises pour contrôler l'accès à vos ressources AWS, veuillez consulter Contrôle de l'accès aux ressources AWS à l'aide de balises dans le Guide de l'utilisateur IAM.

AWS KMSécrit une entrée dans votre AWS CloudTrail journal lorsque vous utilisez les ListResourceTagsopérations TagResourceUntagResource, ou.

Gestion des balises de clé KMS dans la console

Vous pouvez ajouter des balises à une clé KMS lorsque vous créez la clé KMS dans la console AWS KMS. Vous pouvez également utiliser l'onglet Tags (Balises) de la console pour ajouter, modifier et supprimer des balises sur les clés gérées par le client. Pour ajouter, modifier, afficher et supprimer des balises d'une clé KMS, vous devez disposer des autorisations requises. Pour plus de détails, veuillez consulter Contrôle de l'accès aux balises.

Ajouter des balises lors de la création d'une clé KMS

Pour ajouter des balises lors de la création d'une clé KMS dans la console, vous devez disposer de l'autorisation `kms:TagResource` dans une politique IAM, en plus des autorisations requises pour créer des clés KMS et afficher les clés KMS dans la console. Au minimum, l'autorisation doit couvrir toutes les clés KMS du compte et de la région.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client. (Vous ne pouvez pas gérer les balises d'une Clé gérée par AWS.)
4. Choisissez le type de clé, puis choisissez Next (Suivant).
5. Saisissez un alias et une description facultative.
6. Saisissez une clé de balise et une valeur de balise facultative. Pour ajouter des balises supplémentaires, sélectionnez Add tag (Ajouter une balise). Pour supprimer une balise, choisissez Remove (Supprimer). Lorsque vous avez terminé d'étiqueter votre nouvelle clé KMS, cliquez sur Next (Suivant).
7. Terminez la création de votre clé KMS.

Afficher et gérer les balises sur les clés KMS existantes

Pour ajouter, afficher, modifier et supprimer des balises dans la console, vous devez disposer d'autorisations d'étiquetage sur la clé KMS. Vous pouvez obtenir cette autorisation à partir de la politique de clé pour la clé KMS ou, si la politique de clé l'autorise, à partir d'une politique IAM qui inclut la clé KMS. Vous avez besoin de ces autorisations en plus des autorisations pour afficher les clés KMS dans la console.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client. (Vous ne pouvez pas gérer les balises d'une Clé gérée par AWS.)
4. Vous pouvez utiliser le filtre du tableau pour afficher uniquement les clés KMS avec des balises particulières. Pour plus de détails, veuillez consulter Tri et filtrage de vos clés KMS.
5. Sélectionnez la case à cocher en regard de l'alias d'une clé KMS.
6. Sélectionnez Actions de clé, Ajouter ou modifier des balises.
7. Sur la page de détails de la clé KMS, sélectionnez l'onglet Tags (Balises).

   - Pour créer votre première balise, sélectionnez Create tag (Créer une balise), saisissez une clé et une valeur de balise (requis), puis sélectionnez Save (Enregistrer).

     Si vous laissez la valeur de balise vide, la valeur de balise réelle est une chaîne nulle ou vide.
   - Pour ajouter une balise, sélectionnez Edit (Modifier), choisissez Add tag (Ajouter une balise), saisissez une clé et une valeur de balise, puis choisissez Save (Enregistrer).
   - Pour modifier le nom ou la valeur d'une balise, choisissez Modifier, effectuez les modifications nécessaires, puis choisissez Enregistrer.
   - Pour supprimer une balise, choisissez Modifier. Sur la ligne de la balise, choisissez Supprimer, puis choisissez Enregistrer.
8. Choisissez Enregistrer pour enregistrer les modifications.

Gestion des balises de clés KMS avec les opérations API

Vous pouvez utiliser l'API AWS Key Management Service (AWS KMS) pour ajouter, supprimer et répertorier des balises pour les clés KMS que vous gérez. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge. Vous ne pouvez pas étiqueter de Clés gérées par AWS.

Pour ajouter, modifier, afficher et supprimer des balises d'une clé KMS, vous devez disposer des autorisations requises. Pour plus de détails, veuillez consulter Contrôle de l'accès aux balises.

Rubriques

- CreateKey: Ajouter des balises à une nouvelle clé KMS
- TagResource: ajouter ou modifier des balises pour une clé KMS
- ListResourceTags: Récupère les tags d'une clé KMS
- UntagResource: Supprimer les balises d'une clé KMS

CreateKey: Ajouter des balises à une nouvelle clé KMS

Vous pouvez ajouter des balises lorsque vous créez une clé gérée par le client. Pour spécifier les balises, utilisez le `Tags` paramètre de l'CreateKeyopération.

Pour ajouter des balises lors de la création d'une clé KMS, l'appelant doit détenir l'autorisation `kms:TagResource` dans une politique IAM. Au minimum, l'autorisation doit couvrir toutes les clés KMS du compte et de la région. Pour plus de détails, veuillez consulter Contrôle de l'accès aux balises.

La valeur du paramètre `Tags` de `CreateKey` est une collection de paires de clés et de valeurs de balises sensibles à la casse. Chaque balise d'une clé KMS doit avoir un nom de balise différent. La valeur de balise peut être une chaîne vide ou nulle.

Par exemple, la commande AWS CLI suivante crée une clé KMS de chiffrement symétrique avec une balise `Project:Alpha`. Lorsque vous spécifiez plusieurs paires clé-valeur, utilisez un espace pour séparer chaque paire.

```
$ aws kms create-key --tags TagKey=Project,TagValue=Alpha
```

Lorsque cette commande aboutit, elle renvoie un objet `KeyMetadata` contenant des informations sur la nouvelle clé KMS. Cependant, l'objet `KeyMetadata` n'inclut pas les balises. Pour obtenir les balises, utilisez l'ListResourceTagsopération.

TagResource: ajouter ou modifier des balises pour une clé KMS

L'TagResourceopération ajoute une ou plusieurs balises à une clé KMS. Vous ne pouvez pas utiliser cette opération pour ajouter ou modifier des balises dans un autre Compte AWS.

Pour ajouter une balise, spécifiez de nouvelles clé et valeur de balises. Pour modifier une balise, spécifiez une clé de balise existante et une nouvelle valeur de balise. Chaque balise d'une clé KMS doit avoir une clé de balise différente. La valeur de balise peut être une chaîne vide ou nulle.

Par exemple, la commande suivante ajoute les balises **`Purpose`** et **`Department`** à un exemple de clé KMS.

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Pretest TagKey=Department,TagValue=Finance
```

Lorsque cette commande aboutit, elle ne renvoie pas de sortie. Pour afficher les balises d'une clé KMS, utilisez l'ListResourceTagsopération.

Vous pouvez également utiliser TagResource pour modifier la valeur de balise d'une balise existante. Pour remplacer une valeur de balise, spécifiez la même clé de balise avec une valeur différente.

Par exemple, cette commande modifie la valeur de la balise `Purpose` de `Pretest` en `Test`.

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Test
```

ListResourceTags: Récupère les tags d'une clé KMS

L'ListResourceTagsopération obtient les balises d'une clé KMS. Le paramètre `KeyId` est obligatoire. Vous ne pouvez pas utiliser cette opération pour afficher les balises sur les clés KMS dans un autre Compte AWS.

Par exemple, la commande suivante obtient les balises pour un exemple de clé KMS.

```
$ aws kms list-resource-tags --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

  "Truncated": false,
  "Tags": [
      {
        "TagKey": "Project",
        "TagValue": "Alpha"
     },
     {
       "TagKey": "Purpose",
       "TagValue": "Test"
     },
     {
       "TagKey": "Department",
```

```
        "TagValue": "Finance"
      }
  ]
}
```

UntagResource: Supprimer les balises d'une clé KMS

L'UntagResourceopération supprime les balises d'une clé KMS. Pour identifier les balises à supprimer, spécifiez les clés de balise. Vous ne pouvez pas utiliser cette opération pour supprimer des balises des clés KMS dans un autre Compte AWS.

Lorsque l'opération `UntagResource` réussit, elle ne renvoie aucune sortie. En outre, si la clé de balise spécifiée n'est pas trouvée sur la clé KMS, elle ne génère pas d'exception ou ne renvoie pas de réponse. Pour vérifier que l'opération a fonctionné, ListResourceTagsutilisez-la.

Par exemple, cette commande supprime la balise **`Purpose`** et sa valeur de la clé KMS spécifiée.

```
$ aws kms untag-resource --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --tag-keys
 Purpose
```

Contrôle de l'accès aux balises

Pour ajouter, afficher et supprimer des balises, que ce soit dans la console AWS KMS ou à l'aide de l'API, les principaux ont besoin d'autorisations d'étiquetage. Vous pouvez fournir ces autorisations dans les politiques de clé. Vous pouvez également les fournir dans les politiques IAM (y compris les politiques de point de terminaison d'un VPC), mais seulement si la politique de clé le permet. La politique AWSKeyManagementServicePowerUsergérée permet aux principaux d'étiqueter, de débaliser et de répertorier les balises sur toutes les clés KMS auxquelles le compte peut accéder.

Vous pouvez également limiter ces autorisations en utilisant des clés de condition globales AWS pour les balises. DansAWS KMS, ces conditions peuvent contrôler l'accès aux opérations de marquage, telles que TagResourceet UntagResource.

> **Note**
>
> Soyez prudent lorsque vous autorisez les principaux à gérer les balises et les alias. La modification d'une balise ou d'un alias permet d'accorder ou de refuser l'autorisation d'utiliser la clé gérée par le client. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation de balises pour contrôler l'accès aux clés KMS.

Pour obtenir des exemples de politiques et plus d'informations, veuillez consulter Contrôle de l'accès en fonction des clés de balises dans le Guide de l'utilisateur IAM.

Les autorisations de création et de gestion de balises fonctionnent comme suit.

km : TagResource

Autorise les principaux à ajouter ou à modifier des balises. Pour ajouter des balises lors de la création d'une clé KMS, le principal doit disposer d'une autorisation dans une politique IAM qui n'est pas limitée à des clés KMS particulières.

km : ListResourceTags

Permet aux principaux d'afficher les balises sur les clés KMS.

km : UntagResource

Permet aux principaux de supprimer des balises des clés KMS.

Autorisations de balises dans les politiques

Vous pouvez fournir des autorisations d'étiquetage dans une politique de clé ou une politique IAM. Par exemple, l'exemple de politique de clé suivant donne à certains utilisateurs l'autorisation d'étiqueter la clé KMS. Il accorde à tous les utilisateurs qui peuvent endosser les rôles Administrateur ou Développeur d'exemple l'autorisation d'afficher les balises.

```
{
  "Version": "2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow all tagging permissions",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/LeadAdmin",
        "arn:aws:iam::111122223333:user/SupportLead"
```

```
      ]},
      "Action": [
          "kms:TagResource",
          "kms:ListResourceTags",
          "kms:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow roles to view tags",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:role/Administrator",
        "arn:aws:iam::111122223333:role/Developer"
      ]},
      "Action": "kms:ListResourceTags",
      "Resource": "*"
    }
  ]
}
```

Pour accorder aux principaux l'autorisation d'étiquetage sur plusieurs clés KMS, vous pouvez utiliser une politique IAM. Pour que cette politique soit efficace, la politique de clé pour chaque clé KMS doit autoriser le compte à utiliser des politiques IAM pour contrôler l'accès à la clé KMS.

Par exemple, la politique IAM suivante permet aux principaux de créer des clés KMS. Il leur permet également de créer et de gérer des balises sur toutes les clés KMS du compte spécifié. Cette combinaison permet aux principaux d'utiliser le paramètre Tags de l'CreateKeyopération pour ajouter des balises à une clé KMS lors de sa création.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKeys",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyTags",
      "Effect": "Allow",
      "Action": [
```

```
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ListResourceTags"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    }
  ]
}
```

Limitation des autorisations de balises

Vous pouvez limiter les autorisations d'étiquetage en utilisant des conditions de politique. Les conditions de politique suivantes peuvent être appliquées aux autorisations `kms:TagResource` et `kms:UntagResource`. Par exemple, vous pouvez utiliser la condition `aws:RequestTag/`*`tag-key`* pour permettre à un principal d'ajouter uniquement des balises particulières, ou empêcher un principal d'ajouter des balises avec des clés de balise particulières. Sinon, vous pouvez utiliser la condition `kms:KeyOrigin` pour empêcher les principaux d'étiqueter ou de désétiqueter les clés KMS avec des éléments de clé importés.

- lois : RequestTag
- aws :ResourceTag/tag-key (politiques IAM uniquement)
- lois : TagKeys
- km : CallerAccount
- km : KeySpec
- km : KeyUsage
- km : KeyOrigin
- km : ViaService

La bonne pratique à adopter lorsque vous utilisez des balises pour contrôler l'accès aux clés KMS consiste à utiliser la clé de condition `aws:RequestTag/`*`tag-key`* ou `aws:TagKeys` pour déterminer quelles balises (ou clés de balise) sont autorisées.

Par exemple, la politique IAM suivante est similaire à la précédente. Toutefois, cette politique permet aux principaux de créer des balises (`TagResource`) et de supprimer des balises `UntagResource` uniquement pour les balises avec une clé de balise `Project`.

Étant donné que `TagResource` les `UntagResource` demandes peuvent inclure plusieurs balises, vous devez spécifier un opérateur `ForAllValues` ou un `ForAnyValue` ensemble avec la TagKeys

condition aws :. L'opérateur `ForAnyValue` exige qu'au moins l'une des clés de balise dans la demande corresponde à l'une des clés de balise dans la politique. L'opérateur `ForAllValues` exige que toutes les clés de balise dans la demande correspondent à l'une des clés de balise dans la politique. L'`ForAllValues`opérateur renvoie également `true` si la demande ne contient aucune balise, mais TagResource UntagResource échoue si aucune balise n'est spécifiée. Pour plus de détails sur les opérateurs d'ensemble, veuillez consulter Utiliser plusieurs clés et valeurs dans le Guide de l'utilisateur IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKey",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyViewAllTags",
      "Effect": "Allow",
      "Action": "kms:ListResourceTags",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPolicyManageTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
          "ForAllValues:StringEquals": {"aws:TagKeys": "Project"}
      }
    }
  ]
}
```

Utilisation de balises pour contrôler l'accès aux clés KMS

Vous pouvez contrôler l'accès à AWS KMS keys en fonction des balises de la clé KMS. Par exemple, vous pouvez écrire une politique IAM qui permet aux principaux d'activer et de désactiver uniquement

les clés KMS possédant une balise particulière. Vous pouvez également utiliser une politique IAM pour empêcher les principaux d'utiliser des clés KMS dans les opérations de chiffrement, sauf si la clé KMS possède une balise particulière.

Cette fonction fait partie de la prise en charge d'AWS KMS pour le contrôle d'accès basé sur les attributs (ABAC). Pour plus d'informations sur l'utilisation des balises pour contrôler l'accès aux ressources AWS, veuillez consulter Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) pour AWS ? et Contrôle de l'accès aux ressources AWS à l'aide de balises dans le Guide de l'utilisateur IAM. Pour obtenir de l'aide pour résoudre les problèmes d'accès liés à l'ABAC, veuillez consulter Résolution des problèmes liés à l'ABAC pour AWS KMS.

> **Note**
>
> Les modifications d'alias et de balises peuvent prendre jusqu'à cinq minutes pour affecter l'autorisation de clé KMS. Les modifications récentes peuvent être visibles dans les opérations d'API avant qu'elles n'affectent l'autorisation.

AWS KMSprend en charge la clé contextuelle de condition globale aws :ResourceTag/tag-key, qui vous permet de contrôler l'accès aux clés KMS en fonction des balises figurant sur la clé KMS. Étant donné que plusieurs clés KMS peuvent avoir la même balise, cette fonction vous permet d'appliquer l'autorisation à un ensemble sélectionné de clés KMS. Vous pouvez également facilement modifier les clés KMS dans l'ensemble en changeant leurs balises.

Dans AWS KMS, la clé de condition `aws:ResourceTag/`*`tag-key`* est prise en charge uniquement dans les politiques IAM. Il n'est pas pris en charge dans les politiques clés, qui ne s'appliquent qu'à une seule clé KMS, ni dans les opérations qui n'utilisent pas une clé KMS particulière, telles que les ListAliasesopérations ListKeysor.

Le contrôle de l'accès à l'aide de balises offre un moyen simple, évolutif et flexible de gérer les autorisations. Toutefois, s'il n'est pas correctement conçu et géré, il peut autoriser ou refuser l'accès à vos clés KMS par inadvertance. Si vous utilisez des balises pour contrôler l'accès, tenez compte des pratiques suivantes.

- Utilisez des balises pour renforcer la bonne pratique de l'accès le moins privilégié. Accordez uniquement aux principaux IAM les autorisations dont ils ont besoin sur les clés KMS qu'ils doivent utiliser ou gérer. Par exemple, utilisez des balises pour étiqueter les clés KMS utilisées pour un projet. Donnez ensuite à l'équipe de projet l'autorisation d'utiliser uniquement les clés KMS avec la balise de projet.

- Soyez prudent lorsque vous donnez aux principaux les autorisations `kms:TagResource` et `kms:UntagResource` qui leur permettent d'ajouter, de modifier et de supprimer des balises. Lorsque vous utilisez des balises pour contrôler l'accès aux clés KMS, la modification d'une balise peut donner aux principaux l'autorisation d'utiliser des clés KMS qu'ils n'avaient alors pas l'autorisation d'utiliser. Elle peut également refuser l'accès aux clés KMS dont d'autres principaux ont besoin pour réaliser leurs tâches. Les administrateurs de clés qui n'ont pas l'autorisation de modifier les politiques de clé ou de créer des octrois peuvent contrôler l'accès aux clés KMS s'ils sont autorisés à gérer les balises.

  Dans la mesure du possible, utilisez une condition de politique, telle que `aws:RequestTag/`*`tag-key`* ou `aws:TagKeys` pour limiter les autorisations d'étiquetage d'un principal à des balises ou des modèles de balises spécifiques sur des clés KMS particulières.
- Passez en revue les principaux de votre Compte AWS qui disposent actuellement d'autorisations d'étiquetage et de désétiquetage et ajustez-les, si nécessaire. Par exemple, la politique de clé par défaut pour les administrateurs de clés de la console inclut les autorisations `kms:TagResource` et `kms:UntagResource` sur cette clé KMS. Les politiques IAM peuvent autoriser les autorisations d'étiquetage et de désétiquetage sur toutes les clés KMS. Par exemple, la politique AWSKeyManagementServicePowerUsergérée permet aux principaux de baliser, de débaliser et de répertorier les balises sur toutes les clés KMS.
- Avant de définir une politique qui dépend d'une balise, examinez les balises des clés KMS dans votre Compte AWS. Assurez-vous que votre politique s'applique uniquement aux balises que vous avez l'intention d'inclure. Utilisez CloudTrail les journaux et les CloudWatch alarmes pour vous avertir des modifications de balises susceptibles d'affecter l'accès à vos clés KMS.
- Les conditions de politique de balise utilisent la correspondance de modèles ; elles ne sont pas liées à une instance particulière d'une balise. Une politique qui utilise des clés de condition basées sur des balises affecte toutes les balises nouvelles et existantes qui correspondent au modèle. Si vous supprimez et recréez une balise qui correspond à une condition de politique, la condition s'applique à la nouvelle balise, comme elle l'a fait pour l'ancienne.

Prenons l'exemple de la politique IAM suivante : Il permet aux principaux d'appeler les opérations GenerateDataKeyWithoutPlaintextet de déchiffrer uniquement sur les clés KMS de votre compte appartenant à la région Asie-Pacifique (Singapour) et dotées d'un `"Project"="Alpha"` tag. Vous pouvez attacher cette politique à des rôles dans l'exemple de projet Alpha.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

L'exemple de politique IAM suivant autorise les principaux à utiliser n'importe quelle clé KMS dans le compte pour les opérations de chiffrement. Mais il interdit aux principaux d'utiliser ces opérations de chiffrement sur les clés KMS avec une identification `"Type"="Reserved"` ou sans identification `"Type"`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAllowCryptographicOperations",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
```

```
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Type": "Reserved"
        }
      }
    },
    {
      "Sid": "IAMDenyNoTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/Type": "true"
        }
      }
    }
  ]
}
```

Activation et désactivation des clés

Vous pouvez activer et désactiver les clés gérées par les clients. Lorsque vous créez une clé KMS, elle est activée par défaut. Si vous désactivez une clé KMS, elle ne peut être utilisée dans aucune opération de chiffrement jusqu'à sa réactivation.

Étant donné qu'il s'agit d'une action temporaire et réversible facilement, la désactivation d'une clé KMS constitue une alternative sûre à sa suppression, action destructrice et irréversible. Si vous envisagez de supprimer une clé KMS, désactivez-la d'abord et configurez une CloudWatch alarme ou un mécanisme similaire pour être certain de ne jamais avoir à utiliser la clé pour déchiffrer des données chiffrées.

Lorsque vous désactivez une clé KMS, elle devient immédiatement inutilisable (sous réserve d'une éventuelle cohérence). Toutefois, les ressources chiffrées à l'aide de clés de données protégées par la clé KMS ne sont pas affectées tant que la clé KMS n'est pas réutilisée, par exemple pour déchiffrer la clé de données. Ce problème affecte les Services AWS, dont beaucoup utilisent des clés de données pour protéger vos ressources. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

Vous ne pouvez pas activer ou désactiver les Clés gérées par AWS ou les Clés détenues par AWS. Les Clés gérées par AWS sont activées en permanence pour pouvoir être utilisées par des services qui utilisent AWS KMS. Les Clés détenues par AWS sont gérées uniquement par le service qui en est propriétaire.

> **Note**
>
> AWS KMS ne soumet pas les éléments de clé des clés gérées par les clients à la rotation tant qu'elles sont désactivées. Pour plus d'informations, consultez Comment fonctionne la rotation automatique des clés.

Rubriques

- Activation et désactivation des clés KMS (console)
- Activation et désactivation de clés KMS (API AWS KMS)

Activation et désactivation des clés KMS (console)

Vous pouvez utiliser la console AWS KMS pour activer et désactiver les clés gérées par les clients.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Cochez les cases en regard des clés KMS que vous voulez activer ou désactiver.
5. Pour activer une clé KMS, choisissez Key actions (Actions de clé), Enable (Activer). Pour désactiver une clé KMS, choisissez Key actions (Actions de clé), Disable (Désactiver).

Activation et désactivation de clés KMS (API AWS KMS)

L'EnableKeyopération active un désactivéAWS KMS key. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge. Le paramètre `key-id` est obligatoire.

Cette opération ne renvoie aucune sortie. Pour voir l'état de la clé, utilisez l'DescribeKeyopération.

```
$ aws kms enable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

L'DisableKeyopération désactive une clé KMS activée. Le paramètre `key-id` est obligatoire.

```
$ aws kms disable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Cette opération ne renvoie aucune sortie. Pour voir l'état de la clé, utilisez l'DescribeKeyopération et consultez le `Enabled` champ.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "MultiRegion": false,
        "Enabled": false,
        "KeyState": "Disabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333"
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Rotation des AWS KMS keys

Pour créer de nouveaux éléments de chiffrement pour vos clés gérées par le client, vous pouvez créer de nouvelles clés KMS, puis modifier vos applications ou alias pour utiliser les nouvelles clés KMS. Sinon, vous pouvez activer la rotation automatique des clés pour une clé KMS existante.

Lorsque vous activez la rotation automatique des clés pour une clé KMS, AWS KMS génère pour cette dernière de nouveaux éléments de chiffrement chaque année. AWS KMS enregistre à perpétuité toutes les versions précédentes des éléments de chiffrement afin que vous puissiez déchiffrer toutes les données chiffrées avec cette clé KMS. AWS KMS ne supprime pas les éléments de clé ayant fait l'objet d'une rotation tant que vous n'avez pas supprimé la clé KMS. Vous pouvez suivre la rotation du matériel clé pour vos clés KMS sur Amazon CloudWatch etAWS CloudTrail.

Lorsque vous utilisez une clé KMS qui a fait l'objet d'une rotation pour chiffrer des données, AWS KMS utilise les éléments de clé actuels. Lorsque vous utilisez la clé KMS qui a fait l'objet d'une rotation pour déchiffrer le texte chiffré, AWS KMS utilise la version des éléments de clé qui a été utilisée pour les chiffrer. Vous ne pouvez pas demander de version spécifique des éléments de clé. Étant donné que AWS KMS déchiffre de manière transparente avec les éléments de clé appropriés, vous pouvez utiliser en toute sécurité une clé KMS tournée dans les applications et Services AWS sans changement de code.

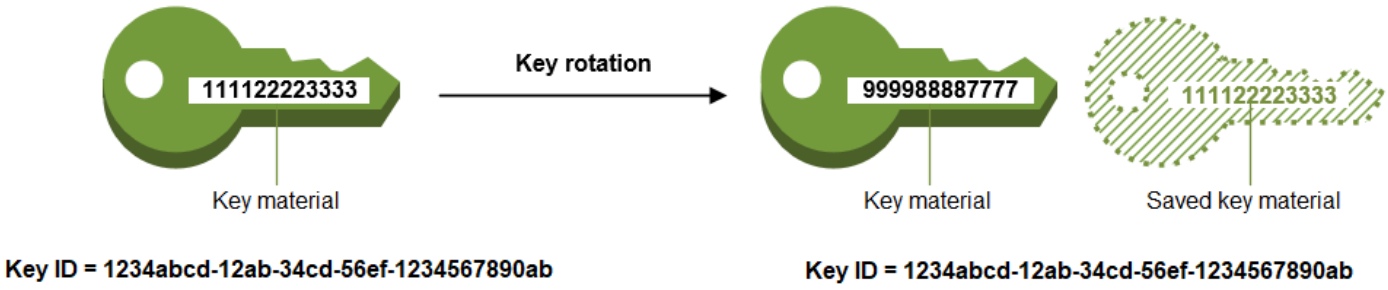
Toutefois, la rotation automatique des clés n'a aucun effet sur les données protégées par la clé KMS. La rotation n'est pas appliquée aux clés de données générées par la clé KMS, les données protégées par la clé KMS ne sont pas rechiffrées et l'effet d'une clé de données compromise n'est pas atténué.

AWS KMS prend en charge la rotation automatique des clés uniquement pour les clés KMS de chiffrement symétriques avec les éléments de clé créés par AWS KMS. La rotation automatique est facultative pour les clés KMS gérées par le client. AWS KMS effectue toujours une rotation des éléments de clé pour les clés KMS gérées par AWS tous les ans. La rotation des clés KMS appartenant à AWS varie.

> **Note**
>
> Intervalle de rotation pour Clés gérées par AWS modifié en mai 2022. Pour plus de détails, consultez Clés gérées par AWS.

La rotation des clés change uniquement les éléments de clé, qui correspondent au secret de chiffrement utilisé dans les opérations de chiffrement. La clé KMS est la même ressource logique,

indépendamment du fait que ses éléments de clé changent ou du nombre de fois où ils changent. Les propriétés de la clé KMS ne changent pas, comme illustré dans l'image suivante.



La rotation automatique des clés offre les avantages suivants :

- Les propriétés de la clé KMS, y compris son ID de clé, son ARN de clé, sa région, ses politiques et ses autorisations, ne changent pas lorsque la clé est l'objet d'une rotation.
- Vous n'avez pas besoin de modifier les applications ou les alias qui font référence à l'ID ou à l'ARN de la clé KMS.
- La rotation des éléments de clé n'affecte pas l'utilisation de la clé KMS dans Service AWS.
- Après avoir activé la rotation des clés, AWS KMS soumet la clé KMS à une rotation automatique chaque année. Vous n'avez pas besoin de vous souvenir de la mise à jour ou de la planifier.

Vous pouvez décider de créer une nouvelle clé KMS et de l'utiliser à la place de la clé KMS d'origine. L'effet est le même que celui obtenu par la rotation des éléments de clé dans une clé KMS existante. Ainsi, on parle souvent à ce sujet de rotation manuelle de la clé. La rotation manuelle est un bon choix lorsque vous souhaitez contrôler la planification de la rotation des clés. Elle permet également de soumettre à une rotation les clés KMS qui ne sont pas éligibles à la rotation automatique des clés, y compris les clés KMS asymétriques, les clés KMS HMAC, les clés KMS dans les magasins de clés personnalisés et les clés KMS avec des éléments de clé importés.

Rotation des clés et tarification

AWS KMS facture des frais mensuels pour chaque version des éléments de clé maintenus pour votre clé KMS. Pour plus d'informations, consultez Tarification AWS Key Management Service.

> **Note**
>
> Vous pouvez utiliser le AWS Cost Explorer Service pour consulter les détails de vos frais de stockage de clés. Par exemple, vous pouvez filtrer votre affichage pour voir le montant total

des frais pour les clés facturées en tant que clés KMS actuelles ou ayant fait l'objet d'une rotation en spécifiant `$REGION-KMS-Keys` pour le type d'utilisation et en regroupant les données par opération d'API.
Vous pouvez toujours voir des instances de l'ancienne opération d'API `Unknown` pour les dates historiques.

Rotation des clés et quotas

Chaque clé KMS compte comme une clé lors du calcul des quotas de ressources de clés, quel que soit le nombre de versions d'éléments de clé qui ont fait l'objet d'une rotation.

Pour plus d'informations sur les éléments de clé et la rotation, veuillez consulter le livre blanc Détails cryptographiques de AWS Key Management Service.

Rubriques

- Pourquoi faire pivoter les clés KMS ?
- Comment fonctionne la rotation automatique des clés
- Activation et désactivation de la rotation automatique des clés
- Rotation manuelle des clés

Pourquoi faire pivoter les clés KMS ?

Les meilleures pratiques cryptographiques découragent la réutilisation intensive des clés qui chiffrent directement les données, telles que les clés de données générées par AWS KMS. Lorsque des clés de données à 256 bits chiffrent des millions de messages, elles peuvent s'épuiser et commencer à produire du texte chiffré avec des motifs subtils que des acteurs intelligents peuvent exploiter pour découvrir les bits contenus dans la clé. Pour éviter cet épuisement des clés, il est préférable d'utiliser les clés de données une seule fois, ou seulement quelques fois, afin de faire pivoter efficacement le contenu clé.

Cependant, les clés KMS sont le plus souvent utilisées comme clés d'encapsulage, également appelées clés de chiffrement. Au lieu de chiffrer les données, les clés d'encapsulage chiffrent les clés de données qui chiffrent vos données. Elles sont donc beaucoup moins souvent utilisées que les clés de données et ne sont presque jamais suffisamment réutilisées pour risquer d'épuiser les clés.

Malgré ce très faible risque d'épuisement, vous devrez peut-être alterner vos clés KMS en raison de règles commerciales ou contractuelles ou de réglementations gouvernementales. Lorsque vous êtes

obligé de faire pivoter les clés KMS, nous vous recommandons d'utiliser la rotation automatique des clés là où elle est prise en charge, et la rotation manuelle des clés lorsque la rotation automatique des clés n'est pas prise en charge.

Comment fonctionne la rotation automatique des clés

La rotation des clés dans AWS KMS est une pratique conçue pour être transparente et facile à utiliser. AWS KMS prend en charge la rotation automatique des clés facultative uniquement pour les clés gérées par le client.

Gestion des éléments de clé

AWS KMS conserve tous les éléments d'une clé KMS, même si la rotation des clés est désactivée. AWS KMS ne supprime les éléments de clé que lorsque vous supprimez la clé KMS.

Utilisation des éléments de clé

Lorsque vous utilisez une clé KMS qui a fait l'objet d'une rotation pour chiffrer des données, AWS KMS utilise les éléments de clé actuels. Lorsque vous utilisez la clé KMS qui a fait l'objet d'une rotation pour déchiffrer le texte chiffré, AWS KMS utilise la même version des éléments de clé qui a été utilisée pour les chiffrer. Vous ne pouvez pas demander de version spécifique des éléments de clé.

Date de rotation

AWS KMS effectue une rotation de l’élément de clé un an (environ 365 jours) après l'activation de la rotation, puis chaque année (environ 365 jours) par la suite.

Clés gérées par le client

La rotation automatique des clés étant facultative sur les clés gérées par le client et pouvant être activée ou désactivée à tout moment, la date de rotation dépend de la date à laquelle la rotation a été activée pour la dernière fois. Cette date peut changer de nombreuses fois tout au long de la durée de vie de la clé.

Par exemple, si vous créez une clé gérée par le client le 1er janvier 2022 et que vous activez la rotation automatique des clés le 15 mars 2022, AWS KMS fait pivoter l’élément de clé le 15 mars 2023, le 15 mars 2024, puis tous les 365 jours par la suite.

Voici des cas particuliers :

- Désactiver la rotation des clés : si vous désactivez la rotation automatique des clés à tout moment, la clé KMS continue d'utiliser la version de l’élément de clé qu'elle utilisait

lorsque la rotation a été désactivée. Si vous activez à nouveau la rotation automatique des clés, AWS KMS soumet l'élément de clé à une rotation un an après la date d'activation de nouvelle rotation et chaque année (environ 365 jours) par la suite.

- Clés KMS désactivées – Lorsqu'une clé KMS est désactivée, AWS KMS ne la soumet pas à une rotation. Toutefois, l'état de rotation de la clé ne change pas et vous ne pouvez pas le modifier tant que la clé KMS est désactivée. Lorsque la clé KMS est réactivée, si les éléments de clé ont plus d'un an, AWS KMS les soumet immédiatement à une rotation, puis chaque année par la suite. Si les éléments de clé ont moins d'un an, AWS KMS reprend la planification initiale de la rotation de la clé.
- Suppression en attente des clés KMS – Tant qu'une clé KMS est en attente de suppression, AWS KMS ne la soumet pas à une rotation. L'état de rotation de la clé est défini sur `false` et vous ne pouvez pas le modifier tant que la suppression est en attente. Si la suppression est annulée, l'état précédent de rotation de la clé est restauré. Si les éléments de clé ont plus d'un an, AWS KMS les soumet immédiatement à une rotation, puis chaque année (environ 365 jours après la dernière rotation) par la suite. Si les éléments de clé ont moins d'un an, AWS KMS reprend la planification initiale de la rotation de la clé.

Clés gérées par AWS

AWS KMS soumet automatiquement les Clés gérées par AWS à une rotation chaque année (environ 365 jours). Vous ne pouvez pas activer ou désactiver la rotation des clés pour Clés gérées par AWS.

L'élément de clé d'un Clé gérée par AWS est d'abord alterné un an après sa date de création, puis chaque année (environ 365 jours après la dernière rotation) par la suite.

> **Note**
>
> En mai 2022, AWS KMS a modifié le calendrier de rotation pour les Clés gérées par AWS de tous les trois ans (environ 1 095 jours) à tous les ans (environ 365 jours).
> Les nouvelles Clés gérées par AWS sont automatiquement soumises à une rotation un an après leur création, puis environ chaque année par la suite.
> Les Clés gérées par AWS existantes sont automatiquement soumises à une rotation un an après leur rotation la plus récente, puis chaque année par la suite.

Clés détenues par AWS

Vous ne pouvez pas activer ou désactiver la rotation des clés pour Clés détenues par AWS. La stratégie de rotation des clés d'une Clé détenue par AWS est déterminée par le service AWS qui crée et gère la clé. Pour plus de détails, reportez-vous à la rubrique Chiffrement au repos dans le Guide de l'utilisateur ou le guide du développeur du service.

Types de clés KMS non pris en charge

La rotation automatique des clés est prise en charge uniquement sur les clés KMS de chiffrement symétriques avec les éléments de clé que AWS KMS génère (Origine = AWS_KMS).

La rotation automatique des clés n'est pas prise en charge sur les types de clés KMS suivants, mais vous pouvez soumettre ces clés KMS à la rotation manuellement.

- Clés KMS asymétriques
- Clés KMS HMAC
- Clés KMS dans des magasins de clés personnalisés
- Clés KMS avec des éléments de clé importés

Clés multi-région

Vous pouvez activer et désactiver la rotation automatique des clés pour les clés multi-région. Vous définissez la propriété uniquement sur la clé principale. Quand AWS KMS synchronise les clés, il copie le paramètre de propriété de la clé principale vers ses clés de réplica. Lorsque les éléments de clé de la clé principale sont soumis à une rotation, AWS KMS les copie automatiquement sur toutes ses clés de réplica. Pour plus de détails, veuillez consulter Rotation de clés multi-région.

Services AWS

Vous pouvez activer la rotation automatique des clés sur les clés gérées par le client que vous utilisez pour le chiffrement côté serveur dans les services AWS. La rotation annuelle est transparente et compatible avec les services AWS.

Surveillance de la rotation des clés

Lorsqu'il fait AWS KMS automatiquement pivoter le contenu clé d'une clé Clé gérée par AWSou d'une clé gérée par le client, il écrit un `KMS CMK Rotation` événement sur Amazon EventBridge et un RotateKey autre dans votre AWS CloudTrail journal. Vous pouvez utiliser ces registres pour vérifier que la clé KMS a fait l'objet d'une rotation.

Cohérence à terme

La rotation automatique des clés est soumise aux mêmes effets de cohérence éventuelle que les autres opérations de gestion AWS KMS. Il peut y avoir un léger retard avant que les nouveaux éléments de clé ne soient disponibles dans AWS KMS. Toutefois, la rotation des éléments de clé n'entraîne aucune interruption ou aucun retard dans les opérations cryptographiques. Les éléments de clé actuels sont utilisés dans les opérations cryptographiques jusqu'à ce que les nouveaux éléments de clé soient disponibles dans AWS KMS. Lorsque les éléments de clé d'une clé multi-région sont automatiquement soumis à une rotation, AWS KMSutilise les éléments de clé actuels jusqu'à ce que les nouveaux éléments soient disponibles dans toutes les régions avec une clé multi-région associée.

Activation et désactivation de la rotation automatique des clés

Les utilisateurs autorisés peuvent utiliser la console AWS KMS et l'API AWS KMS pour activer et désactiver la rotation automatique des clés et afficher l'état de rotation des clés.

Lorsque vous activez la rotation automatique des clés, AWS KMS soumet les éléments de clé de la clé KMS à une rotation un an après la date d'activation, puis chaque année par la suite.

Rubriques

- Activation et désactivation de la rotation des clés (console)
- Activation et désactivation de la rotation de clés (API AWS KMS)

Activation et désactivation de la rotation des clés (console)

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client. (Vous ne pouvez pas activer ou désactiver la rotation des Clés gérées par AWS. Elles sont automatiquement soumises à la rotation tous ans.)
4. Choisissez l'alias ou l'ID d'une clé KMS.
5. Choisissez l'onglet Rotation des clés d'accès.

L'onglet Key rotation (Rotation des clés) apparaît uniquement sur la page détaillée des clés KMS de chiffrement symétriques avec les éléments de clé générés par AWS KMS (l'Origin (Origine) est AWS_KMS), y compris les clés KMS de chiffrement symétriques multi-région.

Vous ne pouvez pas soumettre automatiquement à la rotation les clés KMS asymétriques, les clés KMS HMAC, les clés KMS avec des éléments de clé importés, ou les clés KMS dans les magasins de clé personnalisés. Cependant, vous pouvez les faire pivoter manuellement.

6. Activez ou désactivez la case Automatically rotate this KMS key every year (Effectuer une rotation automatique de cette clé KMS chaque année).

> **Note**
>
> Si une clé KMS est désactivée ou en attente de suppression, la case Automatically rotate this KMS key every year (Effectuer une rotation automatique de cette clé KMS chaque année) est décochée et vous ne pouvez pas la modifier. L'état de rotation des clés est restauré lorsque vous activez la clé KMS ou annulez la suppression. Pour plus de détails, veuillez consulter Comment fonctionne la rotation automatique des clés et Principaux états des clés AWS KMS.

7. Choisissez Enregistrer.

Activation et désactivation de la rotation de clés (API AWS KMS)

Vous pouvez utiliser l'API AWS Key Management Service (AWS KMS) pour activer ou désactiver la rotation automatique des clés et afficher l'état de rotation actuel des clés gérées par les clients. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser tout langage de programmation pris en charge.

L'EnableKeyRotationopération active la rotation automatique des touches pour la clé KMS spécifiée. L'DisableKeyRotationopération le désactive. Pour identifier la clé KMS dans ces opérations, utilisez son ID de clé ou son ARN de clé. Par défaut, la rotation des clés est désactivée pour les clés gérées par le client.

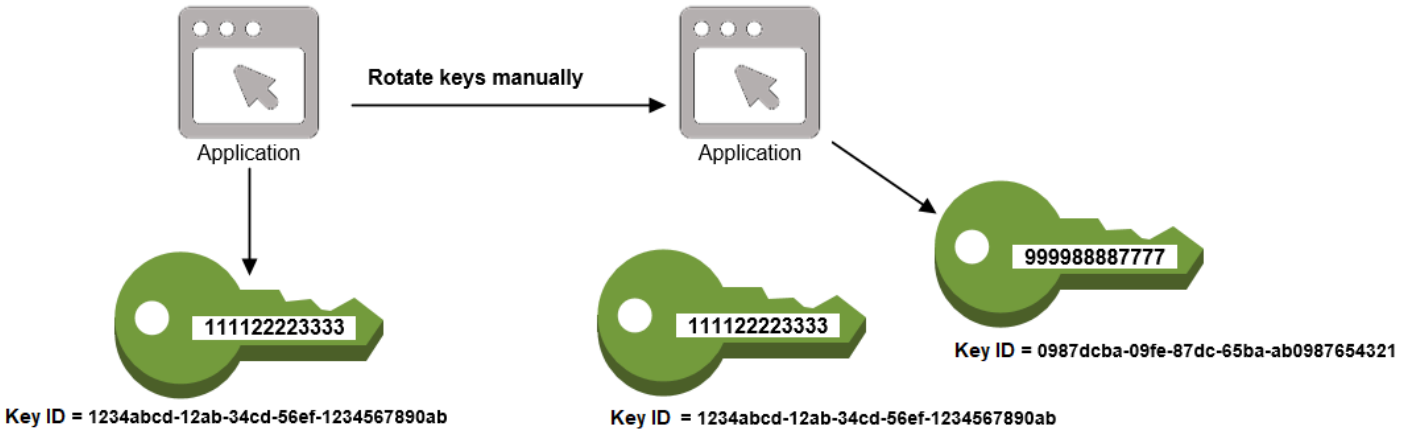
L'exemple suivant active la rotation des clés sur la clé KMS de chiffrement symétrique spécifiée et utilise l'GetKeyRotationStatusopération pour voir le résultat. Ensuite, il désactive la rotation des clés et, à nouveau, utilise GetKeyRotationStatus pour afficher la modification.

```
$ aws kms enable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

```
$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyRotationEnabled": true
}

$ aws kms disable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyRotationEnabled": false
}
```

Rotation manuelle des clés

Vous pouvez vouloir créer une clé KMS et l'utiliser à la place d'une clé KMS actuelle au lieu d'activer la rotation automatique des clés. Lorsque la nouvelle clé KMS possède des éléments de chiffrement différents de ceux de la clé KMS actuelle, l'utilisation de la nouvelle clé KMS a le même effet que la modification des éléments de clé d'une clé KMS existante. Le processus de remplacement d'une cléKMS par une autre est connu sous le nom de rotation manuelle de clés.



Si vous préférez, vous pouvez effectuer une rotation manuelle des clés afin de pouvoir contrôler la fréquence de rotation. Il s'agit également d'une bonne solution pour les clés KMS qui ne sont pas éligibles à la rotation automatique des clés, comme les clés KMS asymétriques, les clés KMS HMAC, les clés KMS des magasins de clés personnalisés ou les clés KMS avec des éléments de clé importés.

> **Note**
>
> Lorsque vous commencez à utiliser la nouvelle clé KMS, assurez-vous de conserver la clé KMS originale activée de telle sorte que AWS KMS puisse déchiffrer les données que la clé KMS originale a chiffrées.

Lorsque vous faites tourner les clés KMS manuellement, vous devez également mettre à jour les références à l'ID ou à l'ARN de la clé KMS dans vos applications. Les alias, qui associent un nom convivial à une clé KMS, facilitent ce processus. Utilisez un alias pour faire référence à une clé KMS dans vos applications. Ensuite, lorsque vous souhaitez modifier la clé KMS que l'application utilise, au lieu de modifier le code de votre application, modifiez la clé KMS cible de l'alias. Pour plus de détails, consultez Utilisation d'alias dans vos applications.

> **Note**
>
> Les alias qui pointent vers la dernière version d'une clé KMS pivotée manuellement constituent une bonne solution pour les DescribeKeyopérations Encrypt,, GenerateDataKeyGenerateDataKeyPairGenerateMac, et Sign. Les alias ne sont pas autorisés dans les opérations qui gèrent les clés KMS, telles que DisableKeyou ScheduleKeyDeletion.
>
> Lorsque vous appelez l'opération Decrypt sur les clés KMS de chiffrement symétriques à rotation manuelle, omettez le paramètre `KeyId` de la commande. AWS KMS utilise automatiquement la clé KMS qui a chiffré le texte chiffré.
>
> Le `KeyId` paramètre est obligatoire lors d'un appel `Decrypt` ou d'une vérification avec une clé KMS asymétrique, ou lors d'un appel VerifyMacavec une clé KMS HMAC. Ces demandes échouent lorsque la valeur de`KeyId`est un alias qui ne pointe plus vers la clé KMS qui a effectué l'opération cryptographique, par exemple lorsqu'une clé fait l'objet d'une rotation manuelle. Pour éviter cette erreur, vous devez spécifier et suivre la clé bonne KMS pour chaque opération.

Pour modifier la clé KMS cible d'un alias, utilisez UpdateAliasl'opération dans l'AWS KMSAPI. Par exemple, cette commande met à jour l'alias `alias/TestKey` pour pointer vers une nouvelle clé KMS. Comme l'opération ne renvoie aucune sortie, l'exemple utilise l'ListAliasesopération pour montrer que l'alias est désormais associé à une autre clé KMS et que le `LastUpdatedDate` champ

est mis à jour. Les ListAliases commandes utilisent le queryparamètre du AWS CLI pour obtenir uniquement l'alias/TestKeyalias.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1521097200.123
        },
    ]
}


$ aws kms update-alias --alias-name alias/TestKey --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321

$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1604958290.722
        },
    ]
}
```

Surveillance des AWS KMS keys

La surveillance est un élément important pour comprendre la disponibilité, l'état et l'utilisation de vos AWS KMS keys dans AWS KMS, et préserver la fiabilité, la disponibilité et les performances de vos solutions AWS. La collecte de données de surveillance à partir de toutes les parties de votre solution AWS vous aidera à résoudre des problèmes de défaillance multipoint, le cas échéant. Toutefois, avant de commencer la surveillance de cos clés KMS, créez un plan de surveillance incluant les réponses aux questions suivantes :

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- À quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

L'étape suivante consiste à contrôler vos clés KMS au fil du temps pour établir une référence d'utilisation normale de AWS KMS et une base pour les attentes dans votre environnement. Lorsque vous contrôlez vos clés KMS, conservez les données d'historique de surveillance pour les comparer aux données actuelles afin d'identifier des modèles normaux et des anomalies, et de concevoir des méthodes pour résoudre les problèmes.

Par exemple, vous pouvez contrôler l'activité d'API AWS KMS et les événements qui affectent vos clés KMS. Lorsque les données dépassent les normes supérieures ou inférieures que vous avez établies, il peut être nécessaire d'enquêter ou de prendre des mesures correctives.

Pour établir une référence pour les modèles normaux, surveillez les éléments suivants :

- Activité d'API AWS KMS pour les opérations de plan de données. Il s'agit d'opérations cryptographiques qui utilisent une clé KMS, telles que Decrypt, Encrypt et ReEncrypt. GenerateDataKey
- Activité d'API AWS KMS pour les opérations de plan de contrôle qui sont importantes pour vous. Ces opérations gèrent une clé KMS, et vous souhaiterez peut-être surveiller celles qui modifient la disponibilité d'une clé KMS (telles que ScheduleKeyDeletionCancelKeyDeletionDisableKey, EnableKey, ImportKeyMaterial, et DeleteImportedKeyMaterial) ou le contrôle d'accès d'une clé KMS (comme PutKeyPolicyet RevokeGrant).
- D'autres métriques AWS KMS (par exemple, le temps restant jusqu'à ce que vos éléments de clé importés expirent) et les événements (tels que l'expiration des éléments de clé importés ou la suppression ou la rotation de clé d'une clé KMS).

Outils de surveillance

AWS fournit différents outils que vous pouvez utiliser pour contrôler vos clés KMS. Vous pouvez configurer certains outils pour qu'ils effectuent la supervision automatiquement, tandis que d'autres

nécessitent une intervention manuelle. Nous vous recommandons d'automatiser le plus possible les tâches de supervision.

Outils de surveillance automatique

Vous pouvez utiliser les outils de surveillance automatique ci-dessous pour contrôler vos clés KMS et signaler un changement éventuel.

- AWS CloudTrailSurveillance des journaux : partagez les fichiers journaux entre les comptes, surveillez les fichiers CloudTrail CloudWatch journaux en temps réel en les envoyant à Logs, rédigez des applications de traitement des journaux avec la bibliothèque de CloudTrail traitement et vérifiez que vos fichiers journaux n'ont pas changé après leur livraison par CloudTrail. Pour plus d'informations, consultez la section Utilisation des fichiers CloudTrail journaux dans le guide de AWS CloudTrail l'utilisateur.
- Amazon CloudWatch Alarms : surveillez une seule métrique sur une période que vous spécifiez et effectuez une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. L'action est une notification envoyée à une rubrique Amazon Simple Notification Service (Amazon SNS) ou à une politique Amazon EC2 Auto Scaling. CloudWatch les alarmes n'appellent pas d'actions simplement parce qu'elles sont dans un état particulier ; l'état doit avoir changé et être maintenu pendant un certain nombre de périodes. Pour plus d’informations, consultez Surveillance avec Amazon CloudWatch.
- Amazon EventBridge — Associez les événements et acheminez-les vers une ou plusieurs fonctions ou flux cibles afin de capturer des informations d'état et, si nécessaire, d'apporter des modifications ou de prendre des mesures correctives. Pour plus d'informations, consultez Surveillance avec Amazon EventBridge le guide de EventBridge l'utilisateur Amazon.
- Amazon CloudWatch Logs — Surveillez, stockez et accédez à vos fichiers journaux depuis AWS CloudTrail ou d'autres sources. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.

Outils de surveillance manuelle

Un autre élément important de la surveillance des clés KMS consiste à surveiller manuellement les éléments non couverts par les CloudWatch alarmes et les événements. Le tableau de bord AWS KMS CloudWatchAWS Trusted Advisor,, et les autres AWS tableaux de bord fournissent une at-a-glance vue de l'état de votre AWS environnement.

Vous pouvez personnaliser les pages Clés gérées par AWS et les clés gérées par le client de la console AWS KMS pour afficher les informations suivantes sur chaque clé KMS :

- ID de clé
- Statut
- Date de création
- Date d'expiration (pour les clés KMS avec des éléments de clé importés)
- Origin
- ID de magasin de clés personnalisé (pour les clés KMS dans des magasins de clés personnalisés)

Le tableau de bord de la console CloudWatch présente les éléments suivants :

- Alarmes et statuts en cours
- Graphiques des alarmes et des ressources
- Statut d'intégrité du service

En outre, vous pouvez utiliser CloudWatch pour effectuer les opérations suivantes :

- Créer des tableaux de bord personnalisés pour surveiller les services de votre choix
- Représenter graphiquement les données de métriques pour résoudre les problèmes et découvrir les tendances
- Rechercher et flotteourir toutes vos métriques de ressources AWS
- Créer et modifier des alarmes pour être informé des problèmes

AWS Trusted Advisor peut vous aider à surveiller vos ressources AWS pour améliorer les performances, la fiabilité, la sécurité et la rentabilité. Quatre contrôles Trusted Advisor sont disponibles pour tous les utilisateurs. Plus de 50 contrôles sont disponibles pour les utilisateurs avec un plan de support Business ou Enterprise. Pour plus d'informations, consultez AWS Trusted Advisor.

Journalisation des appels d'API AWS KMS avec AWS CloudTrail

AWS KMSest intégré à AWS CloudTrailun service qui enregistre tous les appels AWS KMS adressés par les utilisateurs, les rôles et les autres AWS services. CloudTrail capture tous les appels d'API AWS KMS sous forme d'événements, y compris les appels depuis la AWS KMS console, les AWS

KMS API, les AWS CloudFormation modèles, le AWS Command Line Interface (AWS CLI) etAWS Tools for PowerShell.

CloudTrail enregistre toutes les AWS KMS opérations, y compris les opérations en lecture seule, telles que ListAliaseset GetKeyRotationStatus, les opérations qui gèrent les clés KMS, telles que CreateKeyet, et les opérations cryptographiques PutKeyPolicy, telles que GenerateDataKeyet Decrypt. Il enregistre également les opérations internes AWS KMS qui vous concernent, telles que DeleteExpiredKeyMaterialDeleteKey, SynchronizeMultiRegionKey, et RotateKey.

CloudTrail enregistre les opérations réussies et les tentatives d'appels qui ont échoué, par exemple lorsque l'accès à une ressource est refusé à l'appelant. Les opérations intercomptes sur clés KMS sont journalisées à la fois sur le compte appelant et le compte propriétaire de la clé KMS. Toutefois, les demandes entre comptes AWS KMS qui sont rejetées parce que l'accès est refusé sont enregistrées uniquement dans le compte de l'appelant.

Pour des raisons de sécurité, certains champs sont omis des entrées du AWS KMS journal, tels que le `Plaintext` paramètre d'une demande de chiffrement, la réponse à une opération cryptographique GetKeyPolicyou toute autre opération cryptographique. Pour faciliter la recherche d'entrées de CloudTrail journal pour des clés KMS spécifiques, AWS KMS ajoute l'ARN clé de la clé KMS affectée au `responseElements` champ des entrées de journal pour certaines opérations de gestion des AWS KMS clés, même si l'opération d'API ne renvoie pas l'ARN de la clé.

Bien que, par défaut, toutes les AWS KMS actions soient enregistrées en tant qu' CloudTrail événements, vous pouvez AWS KMS les exclure d'un CloudTrail suivi. Pour plus de détails, consultez Exclusion d'événements AWS KMS d'un journal de suivi.

En savoir plus :

- Pour CloudTrail obtenir des exemples d'AWS KMSopérations enregistrées pour une enclave AWS Nitro, consultezDemandes de surveillance pour les enclaves Nitro.

Rubriques

- Enregistrement des événements dans CloudTrail
- Recherche d'événements dans CloudTrail
- Exclusion d'événements AWS KMS d'un journal de suivi
- Exemples d'entrées de journal AWS KMS

Enregistrement des événements dans CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité se produit dansAWS KMS, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements dans votre Compte AWS, y compris les événements pour AWS KMS, créez un journal d'activité. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal de suivi consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez en configurer d'autres Services AWS pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. section withinPour plus d'informations, consultez :

- Présentation de la création d'un journal d'activité
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur. Pour en savoir plus sur les autres façons de contrôler l'utilisation de vos clés KMS, veuillez consulter Surveillance des AWS KMS keys.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification racine ou avec les informations d'identification d'un utilisateur IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été effectuée par un autre Service AWS.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Recherche d'événements dans CloudTrail

Pour rechercher des entrées de CloudTrail journal, utilisez la CloudTrail console ou l'CloudTrail LookupEventsopération. CloudTrail prend en charge de nombreuses valeurs d'attribut pour filtrer votre recherche, notamment le nom de l'événement, le nom d'utilisateur et la source de l'événement.

Pour vous aider à rechercher des entrées de AWS KMS journal dans CloudTrail, AWS KMS remplit les champs d'entrée de CloudTrail journal suivants.

> **Note**
>
> À compter de décembre 2022, AWS KMS remplit les attributs Resource type (Type de ressource) et Resource name (Nom de ressource) dans toutes les opérations de gestion qui modifient une clé KMS particulière. Ces valeurs d'attribut peuvent être nulles dans les anciennes CloudTrail entrées pour les opérations suivantes : CreateAliasCreateGrantDeleteAliasDeleteImportedKeyMaterial, ImportKeyMaterial, ReplicateKey, RetireGrant, RevokeGrant, UpdateAlias, et UpdatePrimaryRegion.

Attribut	Valeur	Entrées de journal
Source de l'événement (`EventSource`)	`kms.amazonaws.com`	Toutes les opérations.
Type de ressource (`ResourceType`)	`AWS::KMS::Key`	Opérations de gestion qui modifient une clé KMS particulière, telles que `CreateKey` et `EnableKey`, mais pas `ListKeys`.
Nom de ressource (`ResourceName`)	ARN de clé (ou ID de clé et ARN de clé)	Opérations de gestion qui modifient une clé KMS particulière, telles que `CreateKey` et `EnableKey`, mais pas `ListKeys`.

Pour vous aider à trouver les entrées de journal pour les opérations de gestion sur des clés KMS particulières, AWS KMS enregistre l'ARN de la clé KMS affectée dans l'élément `responseElements.keyId` de l'entrée de journal, même si l'opération d'API AWS KMS ne renvoie pas l'ARN de la clé.

Par exemple, un appel réussi à l'DisableKeyopération ne renvoie aucune valeur dans la réponse, mais au lieu d'une valeur nulle, la `responseElements.keyId` valeur de l'entrée du DisableKey journal inclut l'ARN de la clé KMS désactivée.

Cette fonctionnalité a été ajoutée en décembre 2022 et concerne les entrées de CloudTrail journal suivantes : CreateAliasCreateGrantDeleteAlias, DeleteKey, DisableKey, EnableKey, EnableKeyRotation, ImportKeyMaterial, RotateKey, SynchronizeMultiRegionKey, TagResource, UntagResource, UpdateAlias, et UpdatePrimaryRegion.

Exclusion d'événements AWS KMS d'un journal de suivi

Pour fournir un enregistrement de l'utilisation et de la gestion de leurs AWS KMS ressources, la plupart des AWS KMS utilisateurs s'appuient sur les événements d'un CloudTrail sentier. Le journal d'activité peut être une source précieuse de données pour l'audit des événements critiques, tels que la création, la désactivation et la suppression de AWS KMS keys, la modification de la politique de clés et l'utilisation de vos clés KMS par les services AWS en votre nom. Dans certains cas, les métadonnées d'une entrée de CloudTrail journal, telles que le contexte de chiffrement d'une opération de chiffrement, peuvent vous aider à éviter ou à résoudre les erreurs.

Cependant, comme AWS KMS peut générer un grand nombre d'événements, AWS CloudTrail vous permet d'exclure des événements AWS KMS d'un journal de suivi. Ce paramètre de journal de suivi exclut tous les événements AWS KMS ; vous ne pouvez pas exclure des événements AWS KMS particuliers.

> **Warning**
>
> L'exclusion d'AWS KMSévénements d'un CloudTrail journal peut masquer les actions qui utilisent vos clés KMS. Soyez prudent lorsque vous accordez aux principaux l'autorisation `cloudtrail:PutEventSelectors` nécessaire pour effectuer cette opération.

Pour exclure des événements AWS KMS d'un journal de suivi :

- Dans la CloudTrail console, utilisez le paramètre des événements du service Log Key Management lorsque vous créez un journal ou que vous le mettez à jour. Pour obtenir des instructions, veuillez

consulter Journalisation des événements de gestion avec la AWS Management Console dans le Guide de l'utilisateur AWS CloudTrail.

- Dans l' CloudTrail API, utilisez l'PutEventSelectorsopération. Ajoutez l'attribut `ExcludeManagementEventSources` à vos sélecteurs d'événements avec la valeur `kms.amazonaws.com`. Pour obtenir un exemple, veuillez consulter Exemple : journal d'activité qui ne consigne pas les événements AWS Key Management Service dans le Guide de l'utilisateur AWS CloudTrail.

Vous pouvez désactiver cette exclusion à tout moment en modifiant le paramètres de la console ou les sélecteurs d'événements d'un journal de suivi. Le journal de suivi recommencera alors à enregistrer les événements AWS KMS. Toutefois, il ne pourra pas récupérer les événements AWS KMS survenus pendant que l'exclusion était effective.

Lorsque vous excluez AWS KMS des événements à l'aide de la console ou de l'API, l'opération CloudTrail `PutEventSelectors` d'API qui en résulte est également enregistrée dans vos CloudTrail journaux. Si AWS KMS les événements n'apparaissent pas dans vos CloudTrail journaux, recherchez un `PutEventSelectors` événement dont l'`ExcludeManagementEventSources`attribut est défini sur`kms.amazonaws.com`.

Exemples d'entrées de journal AWS KMS

AWS KMSécrit des entrées dans votre CloudTrail journal lorsque vous appelez une AWS KMS opération et lorsqu'un AWS service appelle une opération en votre nom. AWS KMSécrit également une entrée lorsqu'il appelle une opération pour vous. Par exemple, il écrit une entrée lorsqu'il supprime une clé KMS dont vous avez programmé la suppression.

Les rubriques suivantes présentent des exemples d'entrées de CloudTrail journal pour les AWS KMS opérations.

Pour des exemples d'entrées de CloudTrail journal de demandes AWS KMS provenant de AWS Nitro Enclaves, consultez. Demandes de surveillance pour les enclaves Nitro

Rubriques

- CancelKeyDeletion
- ConnectCustomKeyStore
- CreateAlias
- CreateCustomKeyStore

- CreateGrant
- CreateKey
- Decrypt
- DeleteAlias
- DeleteCustomKeyStore
- DeleteExpiredKeyMaterial
- DeleteImportedKeyMaterial
- DeleteKey
- DescribeCustomKeyStores
- DescribeKey
- DisableKey
- DisableKeyRotation
- DisconnectCustomKeyStore
- EnableKey
- EnableKeyRotation
- Encrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- GenerateRandom
- GetKeyPolicy
- GetKeyRotationStatus
- GetParametersForImport
- ImportKeyMaterial
- ListAliases
- ListGrants
- PutKeyPolicy
- ReEncrypt

- ReplicateKey
- RetireGrant
- RevokeGrant
- RotateKey
- ScheduleKeyDeletion
- Sign (Signer)
- SynchronizeMultiRegionKey
- TagResource
- UntagResource
- UpdateAlias
- UpdateCustomKeyStore
- UpdateKeyDescription
- UpdatePrimaryRegion
- VerifyMac
- Vérification
- Exemple 1 d'Amazon EC2
- Exemple 2 d'Amazon EC2

CancelKeyDeletion

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération CancelKeyDeletion. Pour plus d'informations sur la suppression de AWS KMS keys, veuillez consulter Suppression de AWS KMS keys.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T21:53:17Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "CancelKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "e3452e68-d4b0-4ec7-a768-7ae96c23764f",
    "eventID": "d818bf03-6655-48e9-8b26-f279a07075fd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ConnectCustomKeyStore

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération `ConnectCustomKeyStore`. Pour plus d'informations sur la connexion d'un magasins de clés personnalisé, veuillez consulter Connecter et déconnecter un magasin de clés AWS CloudHSM.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
```

```
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ConnectCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

CreateAlias

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'CreateAliasopération. L'élément `resources` comprend des champs pour l'alias et les ressources clés KMS. Pour plus d'informations sur la création d'alias dans AWS KMS, veuillez consulter Création d'un alias.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-14T23:08:31Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "aliasName": "alias/ExampleAlias",
        "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "caec1e0c-ce03-419e-bdab-6ab1f7c57c01",
    "eventID": "2dd6e784-8286-46a6-befd-d64e5a02fb28",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


CreateCustomKeyStore

L'exemple suivant illustre une entrée de journal AWS CloudTrail générée en appelant l'opération `CreateCustomKeyStore` sur un magasin de clés AWS CloudHSM. Pour plus d'informations sur les magasins de clés personnalisés, veuillez consulter Créer un magasin de clés AWS CloudHSM.


```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```


CreateGrant

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'CreateGrantopération. Pour plus d'informations sur la création d'octrois dans AWS KMS, veuillez consulter Octrois dans AWS KMS.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.


```
{
  "eventVersion": "1.02",
  "userIdentity": {
      "type": "IAMUser",
```

```
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:53:12Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "constraints": {
          "encryptionContextSubset": {
              "ContextKey1": "Value1"
          }
      },
      "operations": ["Encrypt",
      "RetireGrant"],
      "granteePrincipal": "EX_PRINCIPAL_ID"
  },
  "responseElements": {
      "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
      "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "f3c08808-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "5d529779-2d27-42b5-92da-91aaea1fc4b5",
  "readOnly": false,
  "resources": [{
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```


CreateKey

Ces exemples montrent les entrées du AWS CloudTrail journal de l'CreateKeyopération.

Une entrée de `CreateKey` journal peut résulter d'une `CreateKey` demande ou de l'`CreateKey`opération d'une ReplicateKeydemande.

L'exemple suivant montre une entrée de CloudTrail journal pour une CreateKeyopération qui crée une clé KMS de chiffrement symétrique. Pour plus d'informations sur la création de clés KMS, veuillez consulter Création de clés.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-10T22:38:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "description": "",
        "origin": "EXTERNAL",
        "bypassPolicyLockoutSafetyCheck": false,
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "keySpec": "SYMMETRIC_DEFAULT",
        "keyUsage": "ENCRYPT_DECRYPT"
    },
    "responseElements": {
        "keyMetadata": {
            "AWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Aug 10, 2022, 10:38:27 PM",
            "enabled": false,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "PendingImport",
            "origin": "EXTERNAL",
```

```
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "requestID": "1aef6713-0223-4ff7-9a6d-781360521930",
    "eventID": "36327b37-f4f6-40a9-92ab-48064ec905a2",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

L'exemple suivant montre le CloudTrail journal d'une `CreateKey` opération qui crée une clé KMS de chiffrement symétrique dans un magasin de AWS CloudHSM clés.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-14T17:39:50Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyUsage": "ENCRYPT_DECRYPT",
        "bypassPolicyLockoutSafetyCheck": false,
        "origin": "AWS_CLOUDHSM",
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "customKeyStoreId": "cks-1234567890abcdef0",
        "description": ""
    },
    "responseElements": {
        "keyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "creationDate": "Oct 14, 2021, 5:39:50 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "AWS_CLOUDHSM",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "cloudHsmClusterId": "cluster-1a23b4cdefg",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "additionalEventData": {
        "backingKey": "{\"keyHandle\":\"19\",\"backingKeyId\":\"backing-key-id\"}"
    },
    "requestID": "4f0b185c-588c-4767-9e90-c618f7e13cad",
    "eventID": "c73964b8-703d-49e4-bd9e-f773d0ee1e65",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

L'exemple suivant montre le CloudTrail journal d'une `CreateKey` opération qui crée une clé KMS de chiffrement symétrique dans un magasin de clés externe.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2022-09-07T22:37:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "tags": [],
        "keyUsage": "ENCRYPT_DECRYPT",
        "description": "",
        "origin": "EXTERNAL_KEY_STORE",
        "multiRegion": false,
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "bypassPolicyLockoutSafetyCheck": false,
        "customKeyStoreId": "cks-1234567890abcdef0",
        "xksKeyId": "bb8562717f809024"
    },
    "responseElements": {
        "keyMetadata": {
```

```
            "aWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Dec 7, 2022, 10:37:45 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "EXTERNAL_KEY_STORE",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false,
            "xksKeyConfiguration": {
                "id": "bb8562717f809024"
            }
        }
    },
    "requestID": "ba197c82-3ac7-487a-8ff4-7736bbeb1316",
    "eventID": "838ad5f4-5fdd-4044-afd7-4dbd88c6af56",
    "readOnly": false,
    "resources": [
        {
            "accountId": "227179770375",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:227179770375:key/39c5eb22-
f37c-4956-92ca-89e8f8b57ab2"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Decrypt

Ces exemples montrent des entrées de journal AWS CloudTrail pour l'opération Decrypt.

L'entrée du CloudTrail journal d'une `Decrypt` opération inclut toujours le `encryptionAlgorithm` in `requestParameters` même si l'algorithme de chiffrement n'a pas été spécifié dans la demande. Le texte chiffré de la demande et le texte brut de la réponse sont omis.

Rubriques

- Déchiffrer avec une clé de chiffrement symétrique standard
- Échec lors du déchiffrement avec une clé de chiffrement symétrique standard
- Déchiffrer avec une clé KMS dans un magasin de clés AWS CloudHSM
- Déchiffrer avec une clé KMS dans un magasin de clés externe
- Échec lors du déchiffrement avec une clé KMS dans un magasin de clés externe

Déchiffrer avec une clé de chiffrement symétrique standard

Voici un exemple d'entrée de CloudTrail journal pour une `Decrypt` opération utilisant une clé de chiffrement symétrique standard.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
```

```
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Échec lors du déchiffrement avec une clé de chiffrement symétrique standard

L'exemple d'entrée de CloudTrail journal suivant enregistre l'échec d'une `Decrypt` opération avec une clé KMS de chiffrement symétrique standard. L'exception (`errorCode`) et le message d'erreur (`errorMessage`) sont inclus pour vous aider à résoudre l'erreur.

Dans ce cas, la clé KMS de chiffrement symétrique spécifiée dans la requête `Decrypt` n'était pas la clé KMS de chiffrement symétrique utilisée pour chiffrer les données.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T18:57:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "errorCode": "IncorrectKeyException"
```

```
    "errorMessage": "The key ID in the request does not identify a CMK that can perform
 this operation.",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "22345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Déchiffrer avec une clé KMS dans un magasin de clés AWS CloudHSM

L'exemple d'entrée de CloudTrail journal suivant enregistre une `Decrypt` opération avec une clé KMS dans un magasin de AWS CloudHSM clés. Toutes les entrées de journal des opérations de chiffrement avec une clé KMS dans un magasin de clés personnalisé incluent un champ `additionalEventData` avec `customKeyStoreId`. `additionalEventData` n'est pas spécifié dans la requête.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
```

```
        "userName": "Alice"
    },
    "eventTime": "2021-10-26T23:41:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Development",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "requestID": "e1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "a79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Déchiffrer avec une clé KMS dans un magasin de clés externe

L'exemple d'entrée de CloudTrail journal suivant enregistre une `Decrypt` opération avec une clé KMS dans un magasin de clés externe. Outre `customKeyStoreId`, le champ `additionalEventData` inclut l'ID de clé externe (`XksKeyId`). `additionalEventData` n'est pas spécifié dans la requête.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T00:26:58Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "encryptionContext": {
            "Department": "Engineering",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-9876543210fedcba9",
        "xksKeyId": "abc01234567890fe"
    },
    "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management"
}
```

Échec lors du déchiffrement avec une clé KMS dans un magasin de clés externe

L'exemple d'entrée de CloudTrail journal suivant enregistre l'échec d'une demande d'`Decrypt`opération avec une clé KMS dans un magasin de clés externe. CloudWatch enregistre les demandes qui échouent, en plus des demandes réussies. Lors de l'enregistrement d'un échec, l'entrée du CloudTrail journal inclut l'exception (ErrorCode) et le message d'erreur qui l'accompagne (ErrorMessage).

Si la requête échouée a atteint votre proxy de magasin de clés externe, comme dans cet exemple, vous pouvez utiliser la valeur `requestId` pour associer la requête échouée à une requête correspondante des journaux de votre proxy de magasin de clés externe, si votre proxy les fournit.

Pour obtenir de l'aide concernant les requêtes `Decrypt` provenant de magasins de clés externes, veuillez consulter la rubrique Erreurs de déchiffrement.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T00:26:58Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "errorCode": "KMSInvalidStateException",
    "errorMessage": "The external key store proxy rejected the request because the
 specified ciphertext or additional authenticated data is corrupted, missing, or
 otherwise invalid.",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
```

```
        "encryptionContext": {
            "Department": "Engineering",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-9876543210fedcba9",
        "xksKeyId": "abc01234567890fe"
    },
    "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


DeleteAlias

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'DeleteAliasopération. Pour plus d'informations sur la suppression d'alias, veuillez consulter Suppression d'un alias.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
```

```
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2014-11-04T00:52:27Z"
            }
        }
    },
    "eventTime": "2014-11-04T00:52:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteAlias",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "aliasName": "alias/my_alias"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9542792-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "12f48554-bb04-4991-9cfc-e7e85f68eda0",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-east-1:111122223333:alias/my_alias",
        "accountId": "111122223333"
    },
    {
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DeleteCustomKeyStore

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération `DeleteCustomKeyStore`. Pour plus d'informations sur les magasins de clés personnalisés, veuillez consulter Supprimer un magasin de clés AWS CloudHSM.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```


DeleteExpiredKeyMaterial

Lorsque vous importez du matériel clé dans une AWS KMS key (clé KMS), vous pouvez définir une date et une heure d'expiration pour ce matériel clé. AWS KMSenregistre une entrée dans votre CloudTrail journal lorsque vous importez le matériel clé (avec les paramètres d'expiration) et lorsque vous AWS KMS supprimez le matériel clé expiré. Pour plus d'informations sur la création de clés KMS avec des éléments de clé importés, veuillez consulter Importation de matériel clé pour les AWS KMS clés.

L'exemple suivant illustre une entrée de journal AWS CloudTrail générée lorsque AWS KMS supprime les éléments de clé expirés.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-01T16:00:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteExpiredKeyMaterial",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cfa932fd-0d3a-4a76-a8b8-616863a2b547",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    }
}
```


DeleteImportedKeyMaterial

Si vous importez du matériel clé dans une clé KMS, vous pouvez supprimer le matériel clé importé à tout moment en utilisant cette DeleteImportedKeyMaterialopération. Lorsque vous supprimez des éléments de clé importés d'une clé KMS, l'état de la clé KMS passe à `PendingImport` et la clé KMS ne peut être utilisée dans aucune opération cryptographique. Pour plus de détails, consultez Suppression des éléments de clé importés.

L'exemple suivant illustre une entrée de journal AWS CloudTrail pour l'opération `DeleteImportedKeyMaterial`.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-10-04T21:43:33Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteImportedKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"&example-key-arn-1;"
    },
    "requestID": "dcf0e82f-dad0-4622-a378-a5b964ad42c1",
    "eventID": "2afbb991-c668-4641-8a00-67d62e1fecbd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DeleteKey

L'exemple suivant montre une entrée de journal AWS CloudTrail générée lorsqu'une clé KMS est supprimée. Pour supprimer une clé KMS, vous devez utiliser l'ScheduleKeyDeletionopération. Une fois le délai d'attente spécifié expiré, AWS KMS supprime la clé KMS et enregistre une entrée comme celle-ci dans votre CloudTrail journal pour enregistrer cet événement.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

Pour un exemple de l'entrée du CloudTrail journal de l'`ScheduleKeyDeletion`opération, consultezScheduleKeyDeletion. Pour plus d'informations sur la suppression de clés KMS, veuillez consulter Suppression de AWS KMS keys.

L'exemple d'entrée de CloudTrail journal suivant enregistre une `DeleteKey` opération sur une clé KMS contenant du contenu cléAWS KMS.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-07-31T00:07:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "b25f9cda-74e1-4458-847b-4972a0bf9668",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
```

```
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
     "managementEvent": true,
    "eventCategory": "Management"
}
```

L'entrée de CloudTrail journal suivante enregistre `DeleteKey` l'opération d'une clé KMS dans un magasin de clés AWS CloudHSM personnalisé.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-10-26T23:41:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"keyHandle\":\"01\",\"backingKeyId\":
\"backing-key-id\",\"deletionStatus\":\"SUCCESS\"}]"
    },
    "eventID": "1234585c-4b0c-4340-ab11-662414b79239",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
```

```
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```

DescribeCustomKeyStores

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération `DescribeCustomKeyStores`. Pour plus d'informations sur les magasins de clés personnalisés, veuillez consulter Afficher un magasin de clés AWS CloudHSM.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeCustomKeyStores",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "2ea1735f-628d-43e3-b2ee-486d02913a78",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

DescribeKey

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'DescribeKeyopération. AWS KMSenregistre une entrée similaire à la suivante lorsque vous appelez l'`DescribeKey`opération ou que vous affichez les clés KMS dans la AWS KMS console. Cet appel est le résultat de l'affichage d'une clé dans la console de gestion AWS KMS.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-26T18:01:36Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DisableKey

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'DisableKeyopération. Pour plus d'informations sur l'activation et la désactivation de AWS KMS keys dans AWS KMS, veuillez consulter Activation et désactivation des clés.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisableKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
```

```
}
```

DisableKeyRotation

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération DisableKeyRotation. Pour plus d'informations sur la rotation automatique des clés, consultez Rotation des AWS KMS keys.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:31:39Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "d6a9351a-ed6e-4581-88d1-2a9a8a538497",
    "eventID": "6313164c-83aa-4cc3-9e1a-b7c426f7a5b1",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management"
}
```

DisconnectCustomKeyStore

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération `DisconnectCustomKeyStore`. Pour plus d'informations sur les magasins de clés personnalisés, veuillez consulter Connecter et déconnecter un magasin de clés AWS CloudHSM.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisconnectCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

EnableKey

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'EnableKeyopération. Pour plus d'informations sur l'activation et la désactivation de AWS KMS keys dans AWS KMS, veuillez consulter Activation et désactivation des clés.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:20Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d528a6fb-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "be393928-3629-4370-9634-567f9274d52e",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
```

```
}
```

EnableKeyRotation

L'exemple suivant montre une entrée dans le AWS CloudTrail journal d'un appel à l'EnableKeyRotationopération. Pour un exemple d'entrée de CloudTrail journal écrite lors de la rotation de la clé, consultezRotateKey. Pour plus d'informations sur la rotation de AWS KMS keys, veuillez consulter Rotation des AWS KMS keys.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:41:56Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "81f5b794-452b-4d6a-932b-68c188165273",
    "eventID": "fefc43a7-8e06-419f-bcab-b3bf18d6a401",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Encrypt

L'exemple suivant illustre une entrée de journal AWS CloudTrail pour l'opération Encrypt.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-07-14T20:17:42Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionContext": {
            "Department": "Engineering"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    },
    "responseElements": null,
    "requestID": "f3423043-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "91235988-eb87-476a-ac2c-0cdc244e6dca",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
```

```
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GenerateDataKeyopération.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
```

```
}
```

GenerateDataKeyPair

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GenerateDataKeyPairopération. Cet exemple enregistre une opération qui génère une paire de clés RSA chiffrée sous une AWS KMS key de chiffrement symétrique.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPairWithoutPlaintext

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GenerateDataKeyPairWithoutPlaintextopération. Cet exemple enregistre une opération qui génère une paire de clés RSA qui est chiffrée sous une AWS KMS key de chiffrement symétrique.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPairWithoutPlaintext",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_4096",
        "encryptionContext": {
            "Index": "5"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyWithoutPlaintext

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GenerateDataKeyWithoutPlaintextopération.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "errorCode": "InvalidKeyUsageException",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "d6b8e411-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f7734272-9ec5-4c80-9f36-528ebbe35e4a",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
```

```
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateMac

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GenerateMacopération.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
     },
    "eventTime": "2022-12-23T19:26:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "macAlgorithm": "HMAC_SHA_512",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
```

```
}
```

GenerateRandom

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GenerateRandomopération. Puisque cette opération n'utilise pas de AWS KMS key, le champ `resources` est vide.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GetKeyPolicy

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GetKeyPolicyopération. Pour plus d'informations sur l'affichage de la politique de clé pour une clé KMS, veuillez consulter Affichage d'une politique de clé.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:50:30Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyPolicy",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "policyName": "default"
    },
    "responseElements": null,
    "requestID": "93746dd6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "4aa7e4d5-d047-452a-a5a6-2cce282a7e82",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


GetKeyRotationStatus

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'GetKeyRotationStatusopération. Pour plus d'informations sur la rotation automatique d'éléments de clé d'une clé KMS, veuillez consulter la rubrique Rotation des AWS KMS keys.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
```

```
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:32:11Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyRotationStatus",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "12f9b7e8-49b9-4c1c-a7e3-34ac0cdf0467",
    "eventID": "3d082126-9e7d-4167-8372-a6cfcbed4be6",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


GetParametersForImport

L'exemple suivant montre une entrée de AWS CloudTrail journal générée lorsque vous utilisez l'GetParametersForImportopération. Cette opération renvoie la clé publique et le jeton d'importation que vous utilisez lorsque vous importez des éléments de clé dans une clé KMS. La même CloudTrail entrée est enregistrée lorsque vous utilisez l'`GetParametersForImport`opération ou que vous utilisez la AWS KMS console pour télécharger la clé publique et le jeton d'importation.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
```

```
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:58:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetParametersForImport",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "wrappingAlgorithm": "RSAES_OAEP_SHA_256",
        "wrappingKeySpec": "RSA_2048"
    },
    "responseElements": null,
    "requestID": "b5786406-e3c7-43d6-8d3c-6d5ef96e2278",
    "eventID": "4023e622-0c3e-4324-bdef-7f58193bba87",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ImportKeyMaterial

L'exemple suivant montre une entrée de AWS CloudTrail journal générée lorsque vous utilisez l'ImportKeyMaterialopération. La même CloudTrail entrée est enregistrée lorsque vous utilisez l'`ImportKeyMaterial`opération ou que vous utilisez la AWS KMS console pour importer du matériel clé dans unAWS KMS key.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.


```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-26T00:08:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ImportKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "validTo": "Jan 1, 2021 8:00:00 PM",
        "expirationModel": "KEY_MATERIAL_EXPIRES"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "89e10ee7-a612-414d-95a2-a128346969fd",
    "eventID": "c7abd205-a5a2-4430-bbfa-fc10f3e2d79f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ListAliases

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'ListAliasesopération. Puisque cette opération n'utilise pas d'alias particulier ou de AWS KMS key, le champ `resources` est vide.

Pour plus d'informations sur l'affichage des alias dans AWS KMS, veuillez consulter Affichage des alias.


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:51:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListAliases",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "limit": 5,
        "marker":
 "eyJiIjoiYWxpYXMvZTU0Y2MxOTMtYTMwNC00YzEwLTliZWItYTJjZjA3NjA2OTJhIiwiYSI6ImFsaWFzL2U1NGNjMTkzL
    },
    "responseElements": null,
    "requestID": "bfe6c190-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ListGrants

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'ListGrantopération. Pour plus d'informations sur les octrois dans AWS KMS, veuillez consulter Octrois dans AWS KMS.


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
```

```
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListGrants",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "marker":
 "eyJncmFudElkIjoiMWY4M2U2ZmM0YTY2NDgxYjQ2Yzc4MTdhM2Y4YmQwMDFkZDNiYmQ1MGVlYTMyY2RmOWFiNWY1Nzc1N
\u003d\u003d",
        "limit": 10
    },
    "responseElements": null,
    "requestID": "e5c23960-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "d24380f5-1b20-4253-8e92-dd0492b3bd3d",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


PutKeyPolicy

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération PutKeyPolicy. Pour plus d'informations sur la mise à jour d'une stratégie de clé, consultez Modification d'une politique de clé.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
```

```
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T20:06:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "PutKeyPolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "policyName": "default",
        "policy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-default-1\",\n
 \"Statement\" : [ {\n    \"Sid\" : \"Enable IAM User Permissions\",\n    \"Effect\" :
 \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::111122223333:root
\"\n    },\n    \"Action\" : \"kms:*\",\n    \"Resource\" : \"*\"\n  } ]\n}",
        "bypassPolicyLockoutSafetyCheck": false
    },
    "responseElements": null,
    "requestID": "7bb906fa-dc21-4350-b65c-808ff0f72f55",
    "eventID": "c217db1f-903f-4a2f-8f88-9580182d6313",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


ReEncrypt

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'ReEncryptopération. Le champ `resources` dans cette entrée de journal spécifie deux AWS KMS keys, la clé KMS source et la clé KMS de destination, respectivement.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T23:09:13Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "sourceEncryptionContext": {
            "Project": "Alpha",
            "Department": "Engineering"
        },
        "destinationKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "destinationEncryptionContext": {
            "Level": "3A"
        }
    },
    "responseElements": null,
    "requestID": "03769fd4-acf9-4b33-adf3-2ab8ca73aadf",
    "eventID": "542d9e04-0e8d-4e05-bf4b-4bdeb032e6ec",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ReplicateKey

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération ReplicateKey. Une `ReplicateKey` demande entraîne une `ReplicateKey` opération et une CreateKeyopération.

Pour plus d'informations sur la réplication de clés multi-région, veuillez consulter Création de clés de réplica multi-région.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-11-18T01:29:18Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReplicateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "replicaRegion": "us-west-2",
        "bypassPolicyLockoutSafetyCheck": false,
        "description": ""
    },
    "responseElements": {
        "replicaKeyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Nov 18, 2020, 1:29:18 AM",
            "enabled": false,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Creating",
            "origin": "AWS_KMS",
            "keyManager": "CUSTOMER",
            "keySpec": "SYMMETRIC_DEFAULT",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": true,
            "multiRegionConfiguration": {
                "multiRegionKeyType": "REPLICA",
                "primaryKey": {
                    "arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                    "region": "us-east-1"
                },
                "replicaKeys": [
                    {
                        "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                        "region": "us-west-2"
                    }
                ]
            }
        },
        "replicaPolicy": "{\n  \"Version\":\"2012-10-17\",\n  \"Statement\":[{\n
 \"Effect\":\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:user/
Alice\"},\n    \"Action\":\"kms:*\",\n    \"Resource\":\"*\"\n  }, {\n    \"Effect
\":\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::012345678901:user/Bob\"},\n
    \"Action\":\"kms:CreateGrant\",\n    \"Resource\":\"*\"\n  }, {\n    \"Effect\":
\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::012345678901:user/Charlie\"},\n
  \"Action\":\"kms:Encrypt\",\n    \"Resource\":\"*\"\n}]\n}",
    },
    "requestID": "abcdef68-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "fedcba44-6773-4f96-8763-1993aec9ae6a",
    "readOnly": false,
    "resources": [
        {
```

```
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RetireGrant

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération RetireGrant. Pour plus d'informations concernant le retrait d'octrois, veuillez consulter Retrait et révocation d'octrois.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:39:33Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RetireGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "requestID": "1d274d57-5697-462c-a004-f25fcc29fa26",
    "eventID": "0771bcfb-3e24-4332-9ac8-e1c06563eecf",
    "readOnly": false,
    "resources": [
```

```
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RevokeGrant

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération RevokeGrant. Pour plus d'informations sur la révocation d'octrois, veuillez consulter Retrait et révocation d'octrois.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:35:17Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RevokeGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "responseElements": null,
    "requestID": "59d94c03-c5b7-428d-ae6e-f2c4b47d2917",
    "eventID": "07a23a39-6526-4ae2-b31e-d35fbe9e24ee",
    "readOnly": false,
```

```
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RotateKey

L'exemple suivant illustre une entrée de journal AWS CloudTrail de l'opération qui soumet une AWS KMS key à la rotation. AWS KMS appelle cette opération lorsqu'il est temps de soumettre à la rotation une clé KMS sur laquelle la rotation automatique des clés est activée. Lorsque vous activez la rotation automatique des clés (EnableKeyRotation), AWS KMS fait pivoter la clé KMS 365 jours plus tard et tous les 365 jours par la suite.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

Pour un exemple de l'entrée du CloudTrail journal qui enregistre l'`EnableKeyRotation`opération, consultezEnableKeyRotation. Pour plus d'informations sur la rotation des clés KMS, veuillez consulter Rotation des AWS KMS keys.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-14T01:41:59Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
```

```
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    }
}
```

ScheduleKeyDeletion

Ces exemples montrent les entrées du AWS CloudTrail journal de l'ScheduleKeyDeletionopération.

Pour un exemple d'entrée de CloudTrail journal écrite lorsque la clé est supprimée, consultezDeleteKey. Pour plus d'informations sur la suppression de AWS KMS keys, veuillez consulter Suppression de AWS KMS keys.

L'exemple suivant enregistre une demande `ScheduleKeyDeletion` de clé KMS à région unique.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-03-23T18:58:30Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "pendingWindowInDays": 20,
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyState": "PendingDeletion",
        "deletionDate": "Apr 12, 2021 18:58:30 PM"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

L'exemple suivant enregistre une demande `ScheduleKeyDeletion` de clé KMS multi-région avec des clés de réplica.

Comme AWS KMS ne supprime pas une clé multi-région tant que toutes ses clés de réplica n'ont pas été supprimées, dans le champ `responseElements`, le `keyState` est `PendingReplicaDeletion` et le champ `deletionDate` est omis.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
```

```
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-28T17:59:05Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "pendingWindowInDays": 30,
        "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "keyState": "PendingReplicaDeletion",
        "pendingWindowInDays": 30
    },
    "requestID": "12341411-d846-42a6-a476-b1cbe3011f89",
    "eventID": "abcda5f-396d-494c-9380-0c47860df5f1",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


L'exemple suivant enregistre une demande `ScheduleKeyDeletion` de clé KMS dans un magasin de clés personnalisé AWS CloudHSM.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-26T23:25:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "pendingWindowInDays": 30
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "deletionDate": "Nov 2, 2021, 11:25:25 PM",
        "keyState": "PendingDeletion",
        "pendingWindowInDays": 30
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]"
    },
    "requestID": "abcd9f60-2c9c-4a0b-a456-d5d998f7f321",
    "eventID": "ca01996a-01b0-4edd-bbbb-25d7b6d1a6fa",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management"
}
```

Sign (Signer)

Ces exemples montrent des entrées de journal AWS CloudTrail pour l'opération Sign (Signer).

L'exemple suivant montre une entrée de CloudTrail journal pour une opération de signature qui utilise une clé RSA KMS asymétrique pour générer une signature numérique pour un fichier.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:36:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Sign",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "messageType": "RAW",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
    },
    "responseElements": null,
    "requestID": "8d0b35e0-46cf-48b9-be99-bf2ebc9ab9fb",
    "eventID": "107b3cac-b125-4556-9702-12a2b9afcff7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
```

```
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

SynchronizeMultiRegionKey

L'exemple suivant illustre une entrée de journal AWS CloudTrail générée lorsque AWS KMS synchronise une clé multi-région. La synchronisation implique des appels inter-régions pour copier les propriétés partagées d'une clé primaire multi-région vers ses clés de réplica. AWS KMS synchronise périodiquement les clés multi-région pour s'assurer que toutes les clés multi-région associées ont les mêmes éléments de clé.

L'`resources`élément de l'entrée du CloudTrail journal inclut la clé ARN de la clé primaire multirégionale, y compris saRégion AWS. Les clés de réplica multi-région associées et leurs régions ne sont pas répertoriées dans cette entrée de journal.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-11-18T02:04:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "SynchronizeMultiRegionKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "12345681-de97-42e9-bed0-b02ae1abd8dc",
    "eventID": "abcdec99-2b5c-4670-9521-ddb8f031e146",
    "readOnly": false,
    "resources": [
```

```
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

TagResource

L'exemple suivant montre une entrée de AWS CloudTrail journal d'un appel à l'TagResourceopération visant à ajouter une balise avec une clé de balise `Department` et une valeur de balise de`IT`.

Pour un exemple d'entrée de `UntagResource` CloudTrail journal écrite lors de la rotation de la clé, consultezUntagResource. Pour plus d'informations sur l'étiquetage AWS KMS keys, veuillez consulter Clés de balisage.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tags": [
```

```
            {
                "tagKey": "Department",
                "tagValue": "IT"
            }
        ]
    },
    "responseElements": null,
    "requestID": "b942584a-f77d-4787-9feb-b9c5be6e746d",
    "eventID": "0a091b9b-0df5-4cf9-b667-6f2879532b8f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UntagResource

L'exemple suivant montre une entrée de AWS CloudTrail journal d'un appel à l'UntagResourceopération de suppression d'une balise dont la clé de balise est égale à`Dept`.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

Pour un exemple d'entrée de `TagResource` CloudTrail journal, voirTagResource. Pour plus d'informations sur l'étiquetage AWS KMS keys, veuillez consulter Clés de balisage.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
```

```
    },
    "eventTime": "2020-07-01T21:19:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UntagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tagKeys": [
            "Dept"
        ]
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "cb1d507b-6015-47f4-812b-179713af8068",
    "eventID": "0b00f4b0-036e-411d-aa75-87eb4a35a4b3",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


UpdateAlias

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'UpdateAliasopération. L'élément `resources` comprend des champs pour l'alias et les ressources clés KMS. Pour plus d'informations sur la création d'alias dans AWS KMS, veuillez consulter Création d'un alias.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
         "principalId": "EX_PRINCIPAL_ID",
         "arn": "arn:aws:iam::111122223333:user/Alice",
         "accountId": "111122223333",
         "accessKeyId": "EXAMPLE_KEY_ID",
         "userName": "Alice"
    },
    "eventTime": "2020-11-13T23:18:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "aliasName": "alias/my_alias",
        "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9472f40-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f72d3993-864f-48d6-8f16-e26e1ae8dff0",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:alias/my_alias"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UpdateCustomKeyStore

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération `UpdateCustomKeyStore` pour mettre à jour l'ID de cluster d'un magasin de clés personnalisé. Pour plus d'informations sur les magasins de clés personnalisés, veuillez consulter Modifier les paramètres d'un magasin de clés AWS CloudHSM.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

UpdateKeyDescription

L'exemple suivant montre une entrée de journal AWS CloudTrail générée en appelant l'opération UpdateKeyDescription.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:22:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateKeyDescription",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "description": "New key description"
    },
    "responseElements": null,
    "requestID": "8c3c1f8b-336d-4896-b034-4eb9916bc9b3",
    "eventID": "f5f3d548-2e9e-4658-8427-9dcb5b1ea791",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

UpdatePrimaryRegion

L'exemple suivant montre les entrées de AWS CloudTrail journal générées en appelant l'UpdatePrimaryRegionopération sur une clé multirégionale.

L'`UpdatePrimaryRegion`opération écrit deux entrées de CloudTrail journal : l'une dans la région avec la clé primaire multirégionale qui est convertie en clé de réplique, et l'autre dans la région avec une clé de réplique multirégion convertie en clé primaire.

CloudTrail les entrées du journal pour cette opération enregistrées en décembre 2022 ou après cette date incluent l'ARN de la clé KMS affectée dans la `responseElements.keyId` valeur, même si cette opération ne renvoie pas l'ARN de la clé.

L'exemple suivant montre une entrée de CloudTrail journal pour la `UpdatePrimaryRegion` région où la clé multirégionale est passée d'une clé primaire à une clé de réplique (us-west-2). Le champ `primaryRegion` montre la région qui héberge désormais la clé primaire (ap-northeast-1).

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-03-10T20:23:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdatePrimaryRegion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "primaryRegion": "ap-northeast-1"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
```

```
    "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

L'exemple suivant représente l'entrée de CloudTrail journal pour `UpdatePrimaryRegion` la région où la clé multirégionale est passée d'une clé de réplique à une clé primaire (ap-northeast-1). Cette entrée de journal n'identifie pas la région principale précédente.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "invokedBy": "kms.amazonaws.com"
    },
    "eventTime": "2021-03-10T20:23:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdatePrimaryRegion",
    "awsRegion": "ap-northeast-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "primaryRegion": "ap-northeast-1"
    },
    "responseElements": {
```

```
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "091e6be5-737f-43c6-8431-e3679d6d0619",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

VerifyMac

L'exemple suivant montre une entrée de AWS CloudTrail journal pour l'VerifyMacopération.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
     },
    "eventTime": "2022-03-31T19:25:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "VerifyMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "macAlgorithm": "HMAC_SHA_384",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "f35da560-edff-4d6e-9b40-fb306fa9ef1e",
    "eventID": "6b464487-6dea-44cd-84ad-225d7450c975",
    "readOnly": true,
    "resources": [
        {
           "accountId": "111122223333",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Vérification

Ces exemples montrent des entrées de journal AWS CloudTrail pour l'opération Verify (Vérifier).

L'exemple suivant montre une entrée de CloudTrail journal pour une opération Verify qui utilise une clé RSA KMS asymétrique pour vérifier une signature numérique.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:50:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Verify",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "messageType": "RAW"
    },
    "responseElements": null,
    "requestID": "c73ab82a-af82-4750-ae2c-b6bb790e9c28",
    "eventID": "3b4331cd-5b7b-4de5-bf5f-82ec22f0dac0",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
```

```
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Exemple 1 d'Amazon EC2

L'exemple suivant enregistre un principal IAM qui crée un volume chiffré à l'aide de la clé de volume par défaut dans la console de gestion Amazon EC2.

L'exemple suivant montre une entrée de CloudTrail journal dans laquelle l'utilisateur Alice crée un volume chiffré avec une clé de volume par défaut dans la console de gestion Amazon EC2. L'enregistrement du fichier journal EC2 inclut un champ `volumeId` doté d'une valeur `"vol-13439757"`. L'enregistrement AWS KMS contient un champ `encryptionContext` doté d'une valeur `"aws:ebs:id": "vol-13439757"`. De même, les identificateurs `principalId` et `accountId` entre les deux enregistrements correspondent. Les enregistrements reflètent le fait que la création d'un volume chiffré génère une clé de données qui est utilisée pour chiffrer le contenu du volume.

```
{
  "Records": [
     {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
      "eventTime": "2014-11-05T20:50:18Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "CreateVolume",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
```

```
      "userAgent": "AWS Internal",
      "requestParameters": {
        "size": "10",
        "zone": "us-east-1a",
        "volumeType": "gp2",
        "encrypted": true
      },
      "responseElements": {
        "volumeId": "vol-13439757",
        "size": "10",
        "zone": "us-east-1a",
        "status": "creating",
        "createTime": 1415220618876,
        "volumeType": "gp2",
        "iops": 30,
        "encrypted": true
      },
      "requestID": "1565210e-73d0-4912-854c-b15ed349e526",
      "eventID": "a3447186-135f-4b00-8424-bc41f1a93b4f",
      "eventType": "AwsApiCall",
      "recipientAccountId": "123456789012"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T20:50:19Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "&AWS; Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-13439757"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
```

```
      },
      "responseElements": null,
      "requestID": "create-123456789012-758241111-1415220618",
      "eventID": "4bd2a696-d833-48cc-b72c-05e61b608399",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```

Exemple 2 d'Amazon EC2

Dans l'exemple suivant, un principal IAM exécutant une instance Amazon EC2 crée et monte un volume de données chiffré sous une clé KMS. Cette action génère plusieurs enregistrements de CloudTrail journal.

Lorsque le volume est créé, Amazon EC2, agissant au nom du client, obtient une clé de données chiffrée à partir de AWS KMS (`GenerateDataKeyWithoutPlaintext`). Ensuite, il crée un octroi (`CreateGrant`) qui lui permet de déchiffrer la clé de données. Lorsque le volume est monté, Amazon EC2 appelle AWS KMS pour déchiffrer la clé de données (`Decrypt`).

L'`instanceId` de l'instance Amazon EC2, `"i-81e2f56c"`, s'affiche dans l'événement `RunInstances`. Le même ID d'instance qualifie le `granteePrincipal` de l'octroi créé (`"111122223333:aws:ec2-infrastructure:i-81e2f56c"`) et le rôle supposé qui est le principal dans l'appel `Decrypt` (`"arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c"`).

L'ARN de clé de la clé KMS qui protège le volume de données, `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`, apparaît dans les trois appels AWS KMS (`CreateGrant`, `GenerateDataKeyWithoutPlaintext` et `Decrypt`).

```
{
  "Records": [
    {
```

```
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:27Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "RunInstances",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "instancesSet": {
          "items": [
            {
              "imageId": "ami-b66ed3de",
              "minCount": 1,
              "maxCount": 1
            }
          ]
        },
        "groupSet": {
          "items": [
            {
              "groupId": "sg-98b6e0f2"
            }
          ]
        },
        "instanceType": "m3.medium",
        "blockDeviceMapping": {
          "items": [
            {
              "deviceName": "/dev/xvda",
              "ebs": {
                "volumeSize": 8,
                "deleteOnTermination": true,
                "volumeType": "gp2"
              }
            },
            {
```

```
            "deviceName": "/dev/sdb",
            "ebs": {
              "volumeSize": 8,
              "deleteOnTermination": false,
              "volumeType": "gp2",
              "encrypted": true
            }
          }
        ]
      },
      "monitoring": {
        "enabled": false
      },
      "disableApiTermination": false,
      "instanceInitiatedShutdownBehavior": "stop",
      "clientToken": "XdKUT141516171819",
      "ebsOptimized": false
    },
    "responseElements": {
      "reservationId": "r-5ebc9f74",
      "ownerId": "111122223333",
      "groupSet": {
        "items": [
          {
            "groupId": "sg-98b6e0f2",
            "groupName": "launch-wizard-2"
          }
        ]
      },
      "instancesSet": {
        "items": [
          {
            "instanceId": "i-81e2f56c",
            "imageId": "ami-b66ed3de",
            "instanceState": {
              "code": 0,
              "name": "pending"
            },
            "amiLaunchIndex": 0,
            "productCodes": {

            },
            "instanceType": "m3.medium",
            "launchTime": 1415223328000,
```

```
            "placement": {
              "availabilityZone": "us-east-1a",
              "tenancy": "default"
            },
            "monitoring": {
              "state": "disabled"
            },
            "stateReason": {
              "code": "pending",
              "message": "pending"
            },
            "architecture": "x86_64",
            "rootDeviceType": "ebs",
            "rootDeviceName": "/dev/xvda",
            "blockDeviceMapping": {

            },
            "virtualizationType": "hvm",
            "hypervisor": "xen",
            "clientToken": "XdKUT1415223327917",
            "groupSet": {
              "items": [
                {
                  "groupId": "sg-98b6e0f2",
                  "groupName": "launch-wizard-2"
                }
              ]
            },
            "networkInterfaceSet": {

            },
            "ebsOptimized": false
          }
        ]
      }
    },
    "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
    "eventID": "cd75a605-2fee-4fda-b847-9c3d330ebaae",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.02",
    "userIdentity": {
```

```
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:35Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "CreateGrant",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "constraints": {
          "encryptionContextSubset": {
            "aws:ebs:id": "vol-f67bafb2"
          }
        },
        "granteePrincipal": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      "responseElements": {
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
      },
      "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
      "eventID": "c1ad79e3-0d3f-402a-b119-d5c31d7c6a6c",
      "readOnly": false,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "EX_PRINCIPAL_ID",
```

```
          "arn": "arn:aws:iam::111122223333:user/Alice",
          "accountId": "111122223333",
          "accessKeyId": "EXAMPLE_KEY_ID",
          "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:32Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-111122223333-758247346-1415223332",
      "eventID": "ac3cab10-ce93-4953-9d62-0b6e5cba651d",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
        "arn": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-81e2f56c",
        "accountId": "111122223333",
        "accessKeyId": "",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
```

```
            "creationDate": "2014-11-05T21:35:38Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "111122223333:aws:ec2-infrastructure",
            "arn": "arn:aws:iam::111122223333:role/aws:ec2-infrastructure",
            "accountId": "111122223333",
            "userName": "aws:ec2-infrastructure"
          }
        }
      },
      "eventTime": "2014-11-05T21:35:47Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "Decrypt",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        }
      },
      "responseElements": null,
      "requestID": "b4b27883-6533-11e4-b4d9-751f1761e9e5",
      "eventID": "edb65380-0a3e-4123-bbc8-3d1b7cff49b0",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```


Surveillance avec Amazon CloudWatch

Vous pouvez suivre votre AWS KMS keys utilisation d'Amazon CloudWatch, un AWS service qui collecte et traite les données brutes pour AWS KMS en faire des indicateurs lisibles en temps quasi réel. Ces données sont enregistrées pour une durée de deux semaines pour que vous puissiez

accéder aux informations historiques, et mieux comprendre l'utilisation de vos clés KMS et leur évolution au fil du temps.

Vous pouvez utiliser Amazon CloudWatch pour vous avertir d'événements importants, tels que les suivants.

- Les éléments de clé importés dans une clé KMS approchent de leur date d'expiration.
- Une clé KMS en attente de suppression est toujours utilisée.
- Les éléments de clé dans une clé KMS ont effectué automatiquement une rotation.
- Une clé KMS a été supprimée.

Vous pouvez également créer une CloudWatch alarme Amazon qui vous avertit lorsque le taux de demandes atteint un certain pourcentage de la valeur du quota. Pour plus de détails, consultez Gérer vos taux de demandes AWS KMS d'API à l'aide de Service Quotas et d'Amazon CloudWatch dans le blog sur la AWS sécurité.

Rubriques

- Métriques et dimensions AWS KMS
- Affichage des métriques AWS KMS
- Création d' CloudWatch alarmes pour surveiller les clés KMS

Métriques et dimensions AWS KMS

AWS KMSprédéfinit CloudWatch les métriques Amazon pour vous permettre de surveiller plus facilement les données critiques et de créer des alarmes. Vous pouvez consulter les AWS KMS statistiques à l'aide de l' CloudWatch API AWS Management Console et de l'Amazon.

Cette section répertorie chaque AWS KMS métrique et les dimensions de chaque métrique, et fournit des conseils de base pour créer des CloudWatch alarmes basées sur ces métriques et dimensions.

> **Note**
>
> Nom du groupe de dimensions :
> Pour afficher une métrique dans la CloudWatch console Amazon, dans la section Metrics, sélectionnez le nom du groupe de dimensions. Vous pouvez ensuite filtrer en fonction du

Metric name (Nom de la métrique). Cette rubrique inclut le nom de la métrique et le nom du groupe de dimensions pour chaque métrique AWS KMS.

Rubriques

- SecondsUntilKeyMaterialExpiration
- ExternalKeyStoreThrottle
- XksProxyCertificateDaysToExpire
- XksProxyCredentialAge
- XksProxyErrors
- XksExternalKeyManagerStates
- XksProxyLatency

SecondsUntilKeyMaterialExpiration

Le nombre de secondes restantes avant l'expiration des éléments de clé importés dans une clé KMS. Cette métrique n'est valide que pour les clés KMS avec des éléments de clé importés (dont l'origine des éléments de clé est `EXTERNAL`) et une date d'expiration.

Utilisez cette métrique pour effectuer le suivi du temps restant avant l'expiration de vos éléments de clé importés. Lorsque cette durée tombe en dessous d'un seuil que vous définissez, vous pouvez réimporter les éléments de clé avec une nouvelle date d'expiration. La métrique `SecondsUntilKeyMaterialExpiration` est spécifique à une clé KMS. Vous ne pouvez pas utiliser cette métrique pour surveiller plusieurs clés KMS ou les clés KMS que vous pourriez créer ultérieurement. Pour obtenir de l'aide sur la création CloudWatch d'une alarme pour surveiller cette métrique, consultezCréation d'une CloudWatch alarme en cas d'expiration du matériel clé importé.

`Minimum` est la statistique la plus utile pour cette métrique. Elle indique le plus petit temps restant pour tous les points de données de la période statistique spécifiée. La seule unité valide pour cette métrique est `Seconds`.

Nom du groupe de dimensions : Per-Key Metrics (Métriques par clé)

Dimensions pour **`SecondsUntilKeyMaterialExpiration`**

Dimension	Description ; liée à AWS
KeyId	Valeur pour chaque clé KMS.

ExternalKeyStoreThrottle

Nombre de requêtes d'opérations cryptographiques sur des clés KMS dans chaque magasin de clés externe qu'AWS KMS limite (répond avec une `ThrottlingException`). Cette métrique ne s'applique qu'aux magasins de clés externes.

La `ExternalKeyStoreThrottle` métrique s'applique uniquement aux clés KMS dans un magasin de clés externe et uniquement aux demandes d'opérations cryptographiques et à l'DescribeKeyopération. AWS KMSlimite ces demandes lorsque le taux de demandes dépasse le quota de demandes de stockage de clés personnalisé pour votre magasin de clés externe. Cette métrique n'inclut pas la limitation par votre proxy de magasin de clés externe ou votre gestionnaire de clés externe.

Utilisez cette métrique pour vérifier et ajuster la valeur du quota de demandes du magasin de clés personnalisé. Si cette métrique indique que AWS KMS limite fréquemment vos demandes pour ces clés KMS, vous pouvez envisager de demander une augmentation de la valeur de votre quota de demandes du magasin de clés personnalisé. Si vous avez besoin d'aide, consultez Requesting a quota increase (Demande d'augmentation de quota) dans le Guide de l'utilisateur Service Quotas.

Si vous obtenez très fréquemment des erreurs `KMSInvalidStateException` avec un message expliquant que la requête a été rejetée « en raison d'un taux de requêtes très élevé » ou que la requête a été rejetée « car le proxy de magasin de clés externe n'a pas répondu à temps », cela peut indiquer que votre gestionnaire de clés externe ou votre proxy de magasin de clés externe ne peut pas suivre le rythme du taux de requêtes actuel. Si possible, réduisez votre taux de requêtes. Vous pouvez également envisager de demander une diminution de la valeur de votre quota de requêtes du magasin de clés personnalisé. La diminution de cette valeur de quota peut augmenter la limitation (et la valeur de la métrique `ExternalKeyStoreThrottle`), mais cela indique qu'AWS KMS rejette rapidement les requêtes excédentaires avant qu'elles ne soient envoyées à votre proxy de magasin de clés externe ou à votre gestionnaire de clés externe. Pour solliciter une réduction de quota, accédez au Centre AWS Support et créez une demande.

Nom du groupe de dimensions : Keystore Throttle Metrics (Métriques de limitation du magasin de clés)

Dimension	Description
CustomKeyStoreId	Valeur pour chaque magasin de clés externe.
KmsOperation	Valeur pour chaque opération d'API AWS KMS. Cette métrique s'applique uniquement aux opérations cryptographiques et à l'opération `DescribeKey` sur les clés KMS dans un magasin de clés externe.
KeySpec	Valeur pour chaque type de clé KMS. La seule spécification de clé prise en charge pour les clés KMS dans un magasin de clés externe est SYMMETRIC_DEFAULT.

XksProxyCertificateDaysToExpire

Nombre de jours avant l'expiration du certificat TLS de votre point de terminaison du proxy de magasin de clés externe (`XksProxyUriEndpoint`). Cette métrique ne s'applique qu'aux magasins de clés externes.

Utilisez cette métrique pour créer une CloudWatch alarme qui vous avertit de l'expiration prochaine de votre certificat TLS. Lorsque le certificat expire, AWS KMS ne peut plus communiquer avec le proxy de magasin de clés externe. Toutes les données protégées par des clés KMS dans votre magasin de clés externe deviennent inaccessibles jusqu'à ce que vous renouveliez le certificat.

Une alerte de certificat informe de l'expiration d'un certificat qui pourrait vous empêcher d'accéder à vos ressources chiffrées. Réglez l'alerte pour donner à votre organisation le temps de renouveler le certificat avant qu'il n'expire.

Nom du groupe de dimensions : XKS Proxy Certificate Metrics (Métriques du certificat du proxy XKS)

Dimension	Description
CustomKeyStoreId	Valeur pour chaque magasin de clés externe.
CertificateName	Nom du sujet (CN) dans le certificat TLS.

XksProxyCredentialAge

Nombre de jours écoulés depuis que les informations d'identification pour l'authentification de proxy (`XksProxyAuthenticationCredential`) du magasin de clés externe actuel ont été associées au magasin de clés externe. Ce décompte commence lorsque vous saisissez les informations d'identification pour l'authentification dans le cadre de la création ou de la mise à jour de votre magasin de clés externe. Cette métrique ne s'applique qu'aux magasins de clés externes.

Cette valeur est conçue pour vous rappeler l'âge de vos informations d'identification pour l'authentification. Toutefois, étant donné que nous commençons le décompte lorsque vous associez les informations d'identification à votre magasin de clés externe, et non lorsque vous créez vos informations d'identification pour l'authentification sur le proxy de votre magasin de clés externe, cela peut ne pas être un indicateur précis de l'âge des informations d'identification sur le proxy.

Utilisez cette métrique pour créer une CloudWatch alarme qui vous rappelle de changer vos informations d'identification d'authentification du proxy de stockage de clés externe.

Nom du groupe de dimensions : Per-Keystore Metrics (Métriques par magasin de clés)

Dimension	Description
CustomKeyStoreId	Valeur pour chaque magasin de clés externe.

XksProxyErrors

Nombre d'exceptions liées à des requêtes AWS KMS adressées à votre proxy de magasin de clés externe. Ce décompte comprend les exceptions que le proxy du magasin de clés externe renvoie à AWS KMS et les erreurs de délai d'expiration qui se produisent lorsque le proxy du magasin de clés externe ne répond pas à AWS KMS dans l'intervalle de délai d'expiration de 250 millisecondes. Cette métrique ne s'applique qu'aux magasins de clés externes.

Utilisez cette métrique pour effectuer le suivi du taux d'erreurs des clés KMS dans votre magasin de clés externe. Elle révèle les erreurs les plus fréquentes, ce qui vous permet de hiérarchiser vos efforts d'ingénierie. Par exemple, les clés KMS qui génèrent des taux élevés d'erreurs non récupérables peuvent indiquer un problème de configuration de votre magasin de clés externe. Pour consulter la configuration de votre magasin de clés externe, veuillez consulter la rubrique Afficher un magasin de clés externe. Pour modifier les paramètres de votre clé externe, veuillez consulter la rubrique Modifier les propriétés du magasin de clés externe.

Nom du groupe de dimensions : XKS Proxy Error Metrics (Métriques d'erreurs du proxy XKS)

Dimension	Description
CustomKeyStoreId	Valeur pour chaque magasin de clés externe.
KmsOperation	Valeur pour chaque opération d'API AWS KMS qui a généré une requête au proxy XKS.
XksOperation	Valeur pour chaque opération de l'API du proxy de magasin de clés externe.
KeySpec	Valeur pour chaque type de clé KMS. La seule spécification de clé prise en charge pour les clés KMS dans un magasin de clés externe est SYMMETRIC_DEFAULT.
ErrorType	Valeurs : • Erreurs récupérables : susceptibles d'être transitoires, telles que des erreurs de mise en réseau. • Erreurs non récupérables : susceptibles d'indiquer un problème avec la configuration du magasin de clés personnalisé ou des composants externes. • N/A : requête réussie ; aucune erreur
Exception Name	Valeurs : • Nom de l'exception • Aucune : requête réussie ; aucune erreur

XksExternalKeyManagerStates

Décompte du nombre d'instances de gestionnaire de clés externe dans chacun des états suivants : `Active`, `Degraded` et `Unavailable`. Les informations relatives à cette métrique proviennent du proxy de magasin de clés externe associé à chaque magasin de clés externe. Cette métrique ne s'applique qu'aux magasins de clés externes.

Les états des instances de gestionnaire de clés externe associées à un magasin de clés externe sont les suivants. Chaque proxy de magasin de clés externe peut utiliser des indicateurs différents

pour mesurer l'état de votre gestionnaire de clés externe. Pour plus de détails, veuillez consulter la documentation de votre proxy de magasin de clés externe.

- `Active` : le gestionnaire de clés externe est sain.
- `Degraded` : le gestionnaire de clés externe est défectueux, mais il peut toujours traiter le trafic.
- `Unavailable` : le gestionnaire de clés externe ne peut pas traiter le trafic.

Utilisez cette métrique pour créer une CloudWatch alarme qui vous avertit en cas de dégradation ou d'indisponibilité d'instances de gestionnaire de clés externes. Pour déterminer quelles instances de gestionnaire de clés externe se trouvent dans chaque état, consultez les journaux du proxy de votre magasin de clés externe.

Nom du groupe de dimensions : XKS External Key Manager Metrics (Métriques du gestionnaire de clés externe XKS)

Dimension	Description
CustomKeyStoreId	Valeur pour chaque magasin de clés externe.
XksExternalKeyManagerState	Valeur pour chaque état.

XksProxyLatency

Nombre de millisecondes nécessaires à un proxy de magasin de clés externe pour répondre à une requête AWS KMS. Si le délai de la requête a expiré, la valeur enregistrée est la limite de délai d'expiration de 250 millisecondes. Cette métrique ne s'applique qu'aux magasins de clés externes.

Utilisez cette métrique pour évaluer les performances de votre proxy de magasin de clés externe et de votre gestionnaire de clés externe. Par exemple, si le proxy expire fréquemment lors des opérations de chiffrement et de déchiffrement, consultez votre administrateur de proxy externe.

Des réponses lentes peuvent également indiquer que votre gestionnaire de clés externe ne peut pas gérer le trafic de requête actuel. AWS KMS recommande que votre gestionnaire de clés externe soit capable de traiter jusqu'à 1 800 requêtes d'opérations cryptographiques par seconde. Si votre

gestionnaire de clés externe ne peut pas gérer le taux de 1 800 requêtes par seconde, pensez à demander une diminution de votre quota de requêtes de clés KMS dans un magasin de clés personnalisé. Les requêtes d'opérations cryptographiques utilisant les clés KMS de votre magasin de clés externe échoueront rapidement, avec une exception de limitation, au lieu d'être traitées puis rejetées par le proxy de votre magasin de clés externe ou le gestionnaire de clés externe.

Nom du groupe de dimensions : XKS Proxy Latency Metrics (Métriques de latence de proxy XKS)

Dimension	Description
CustomKeyStoreId	Valeur pour chaque magasin de clés externe.
KmsOperation	Valeur pour chaque opération d'API AWS KMS qui a généré une requête au proxy XKS.
XksOperation	Valeur pour chaque opération de l'API du proxy de magasin de clés externe.
KeySpec	Valeur pour chaque type de clé KMS. La seule spécification de clé prise en charge pour les clés KMS dans un magasin de clés externe est SYMMETRIC_DEFAULT.

Affichage des métriques AWS KMS

Vous pouvez consulter les AWS KMS statistiques à l'aide de l' CloudWatch API AWS Management Console et de l'Amazon.

Pour afficher les métriques à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, changez la région. Dans la barre de navigation, sélectionnez la région où résident vos ressources AWS.
3. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
4. Sous l'onglet Parcourir, recherchez `KMS` et choisissez KMS.
5. Choisissez le nom du groupe de dimensions de la métrique que vous souhaitez consulter.

   Par exemple, pour la métrique `SecondsUntilKeyMaterialExpiration`, choisissez Per-Key Metrics (Métriques par clé).

6. Pour un graphique de la valeur de métrique, choisissez le nom de la métrique, puis choisissez `Add to graph`. Pour convertir le graphique linéaire en valeur, choisissez Ligne, puis choisissez Numéro.

Pour consulter les statistiques à l'aide de l' CloudWatch API Amazon

Pour consulter AWS KMS les métriques à l'aide de l' CloudWatch API, envoyez une ListMetricsdemande avec `Namespace` set to`AWS/KMS`. L'exemple suivant montre comment procéder avec l'AWS Command Line Interface (AWS CLI).

```
$  aws cloudwatch list-metrics --namespace AWS/KMS

{
    "Metrics": [
        {
            "Namespace": "AWS/KMS",
            "MetricName": "SecondsUntilKeyMaterialExpiration",
            "Dimensions": [
                {
                    "Name": "KeyId",
                    "Value": "1234abcd-12ab-34cd-56ef-1234567890ab"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "ExtenalKeyStoreThrottle",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Encrypt"
                },
                {
                    "Name": "KeySpec",
                    "Value": "SYMMETRIC_DEFAULT"
                }
            ]
        },
```

```
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyCertificateDaysToExpire",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "CertificateName",
                    "Value": "myproxy.xks.example.com"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyCredentialAge",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyErrors",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Decrypt"
                },
                {
                    "Name": "XksOperation",
                    "Value": "Decrypt"
                },
                {
                     "Name": "KeySpec",
                     "Value": "SYMMETRIC_DEFAULT"
                },
```

```
                {
                    "Name": "ErrorType",
                    "Value": "Retryable errors"
                },
                {
                     "Name": "ExceptionName",
                     "Value": "KMSInvalidStateException"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyHsmStates",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "XksProxyHsmState",
                    "Value": "Active"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyLatency",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Decrypt"
                },
                {
                    "Name": "XksOperation",
                    "Value": "Decrypt"
                },
                {
                     "Name": "KeySpec",
                     "Value": "SYMMETRIC_DEFAULT"
                }
```

```
            ]
        }
    ]
}
```

Création d' CloudWatch alarmes pour surveiller les clés KMS

Vous pouvez créer une CloudWatch alarme Amazon en fonction d'une AWS KMS métrique. L'alarme envoie un message électronique lorsqu'une valeur de métrique dépasse un seuil spécifié dans la configuration de l'alarme. L'alarme peut envoyer le message électronique à une rubrique Amazon Simple Notification Service (Amazon SNS) ou à une stratégie Amazon EC2 Auto Scaling. Pour obtenir des informations détaillées sur les CloudWatch alarmes, consultez la section Utilisation des CloudWatch alarmes Amazon dans le guide de CloudWatch l'utilisateur Amazon

Création d'une alarme en cas d'expiration des éléments de clé importés

Vous pouvez utiliser la `SecondsUntilKeyMaterialExpiration`métrique pour créer une CloudWatch alarme qui vous avertit lorsque le contenu clé importé d'une clé KMS est sur le point d'expirer.

Lorsque vous importez des éléments de clé dans une clé KMS, vous pouvez éventuellement spécifier une date et heure à laquelle les éléments de clé doivent expirer. Lorsque les éléments de clé expirent, AWS KMS supprime les éléments de clé et la clé KMS devient inutilisable. Pour utiliser la clé KMS à nouveau, vous devez réimporter les éléments de clé.

Pour obtenir des instructions, veuillez consulter Création d'une CloudWatch alarme en cas d'expiration du matériel clé importé.

Création d'une alarme pour les clés KMS en attente de suppression

Lorsque vous planifiez la suppression d'une clé KMS, AWS KMS applique une période d'attente avant de supprimer la clé KMS. Vous pouvez utiliser la période d'attente pour vous assurer de ne pas avoir besoin de la clé KMS maintenant ni par la suite. Vous pouvez également configurer une CloudWatch alarme pour vous avertir si une personne ou une application tente d'utiliser la clé KMS lors d'une opération cryptographique pendant la période d'attente. Si vous recevez une notification de ce type d'alarme, vous pouvez annuler la suppression de la clé KMS.

Pour obtenir des instructions, veuillez consulter Création d'une alarme qui détecte l'utilisation d'une clé KMS en attente de suppression.

Créer une alerte pour surveiller un magasin de clés externe

Vous pouvez créer des CloudWatch alarmes en fonction des métriques relatives aux banques de clés externes et aux clés KMS des banques de clés externes.

Par exemple, nous vous recommandons de définir une CloudWatch alarme pour vous avertir lorsque le certificat TLS de votre banque de clés externe est sur le point d'expirer (XksProxyCertificateDaysToExpire), lorsque votre proxy de magasin de clés externe signale que vos instances de gestionnaire de clés externe sont dégradées ou indisponibles (XksProxyHsmStates).

Pour obtenir des instructions, consultez Surveiller un magasin de clés externe.

Surveillance avec Amazon EventBridge

Vous pouvez utiliser Amazon EventBridge (anciennement Amazon CloudWatch Events) pour vous avertir des événements importants suivants survenus dans le cycle de vie de vos clés KMS.

- Les éléments de clé dans une clé KMS ont effectué automatiquement une rotation.
- Le matériel clé importé dans la clé KMS expirée
- Une clé KMS dont la suppression avait été planifiée a été supprimée.

AWS KMSs'intègre EventBridge à Amazon pour vous informer des événements importants qui affectent vos clés KMS. Chaque événement est représenté au format JSON (JavaScriptObject Notation) et inclut le nom de l'événement, la date et l'heure auxquelles l'événement s'est produit, ainsi que les événements concernés. Vous pouvez collecter ces évènements et définir des règles qui les acheminent vers une ou plusieurs cibles, comme des fonctions AWS Lambda, des rubriques Amazon SNS, des files d'attente Amazon SQS, des flux dans Amazon Kinesis Data Streams ou des cibles intégrées.

Pour plus d'informations sur l'utilisation EventBridge avec d'autres types d'événements, notamment ceux émis AWS CloudTrail lors de l'enregistrement d'une demande d'API en lecture/écriture, consultez le guide de EventBridge l'utilisateur Amazon.

Les rubriques suivantes décrivent les EventBridge événements AWS KMS générés.

Rotation de clé CMK dans KMS

AWS KMS prend en charge la rotation automatique de l'élément clé dans des clés KMS symétriques. de chiffrement La rotation annuelle des éléments de clé est facultative pour les clés gérées par le

client. Les éléments de clé pour les Clés gérées par AWS font l'objet d'une rotation automatique chaque année.

Chaque fois qu'il AWS KMS fait pivoter un matériau clé, il envoie un `KMS CMK Rotation` événement à EventBridge. AWS KMSgénère cet événement dans la mesure du possible.

Voici un exemple de cet événement.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Expiration d'éléments de clé importés KMS

Lorsque vous importez des éléments de clé dans une clé KMS, vous pouvez éventuellement spécifier une heure à laquelle les éléments de clé doivent expirer. Lorsque le contenu clé expire, le AWS KMS supprime et envoie un `KMS Imported Key Material Expiration` événement correspondant à EventBridge. AWS KMSgénère cet événement dans la mesure du possible.

Voici un exemple de cet événement.

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
```

```
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Suppression d'une clé CMK dans KMS

Lorsque vous planifiez la suppression d'une clé KMS, AWS KMS applique une période d'attente avant de supprimer la clé KMS. Une fois la période d'attente terminée, AWS KMS supprime la clé KMS et envoie un `KMS CMK Deletion` événement à EventBridge. AWS KMSgarantit cet EventBridge événement. En raison de nouvelles tentatives, il peut générer plusieurs événements en quelques secondes qui suppriment la même clé KMS.

Voici un exemple de cet événement.

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Création de ressources AWS KMS avec AWS CloudFormation

AWS Key Management Service est intégré à AWS CloudFormation, un service qui vous aide à modéliser et à configurer vos ressources AWS afin que vous puissiez consacrer moins de temps à la création et à la gestion de vos ressources et de votre infrastructure. Vous créez un modèle qui décrit les clés et les alias KMS, puis AWS CloudFormation alloue et configure ces ressources pour vous. Pour plus d'informations sur la AWS KMS prise en charge de CloudFormation, consultez la référence au type de ressource KMS dans le guide de AWS CloudFormation l'utilisateur.

Lorsque vous utilisez AWS CloudFormation, vous pouvez réutiliser votre modèle pour configurer vos ressources AWS KMS de manière cohérente et répétée. Décrivez vos ressources une seule fois, puis allouez-les autant de fois que vous le souhaitez dans plusieurs Comptes AWS et régions.

Pour allouer et configurer des ressources pour AWS KMS et d'autres services AWS, vous devez comprendre les modèles AWS CloudFormation. Les modèles sont des fichiers texte formatés en JSON ou YAML. Ces modèles décrivent les ressources que vous souhaitez allouer dans vos piles AWS CloudFormation. Si JSON ou YAML ne vous est pas familier, vous pouvez utiliser AWS CloudFormation Designer pour vous aider à démarrer avec des modèles AWS CloudFormation. Pour plus d'informations, consultez Qu'est-ce que AWS CloudFormation Designer ? dans le AWS CloudFormationGuide de l'utilisateur.

Régions

AWS KMS CloudFormation les ressources sont prises en charge dans toutes les régions où elles AWS CloudFormation sont prises en charge.

Ressources AWS KMS dans les modèles AWS CloudFormation

AWS KMS prend en charge les ressources AWS CloudFormation suivantes.

- `AWS::KMS::Key` crée une clé KMS symétrique ou asymétrique. Vous pouvez utiliser cette ressource pour créer une clé KMS primaire multi-région symétrique ou asymétrique. Pour créer une clé de réplica multi-région, utilisez la ressource `AWS::KMS::ReplicaKey`. Vous ne pouvez pas utiliser cette ressource pour créer des clés KMS avec des éléments de clé importés ou des clés KMS dans un magasin de clés personnalisé.
- `AWS::KMS::Alias` crée un alias et l'associe à une clé KMS. La clé KMS peut être définie dans le modèle ou créée par un autre mécanisme.
- `AWS::KMS::ReplicaKey` crée une clé de réplica multi-région. Pour créer une clé principale multi-région, utilisez la ressource `AWS::KMS::Key`. Vous ne pouvez pas utiliser cette ressource pour répliquer des clés multi-régions avec des éléments de clé importés. Pour plus de détails sur les clés multi-région, veuillez consulter Clés multi-régions dans AWS KMS.

> **Important**
>
> Si vous modifiez la valeur d'une propriété `KeyUsage`, `KeySpec` ou `MultiRegion`, sur une KMS existante, la clé KMS existante est planifiée pour la suppression et une nouvelle clé KMS est créée avec la valeur spécifiée.

La clé KMS existante devient inutilisable si sa suppression est planifiée. Si vous n'annulez pas la suppression planifiée de la clé KMS existante en dehors de AWS CloudFormation, toutes les données chiffrées sous la clé KMS existante deviennent irrécupérables lorsque la clé KMS est supprimée.

Les clés KMS créées par le modèle sont des ressources réelles dans votre Compte AWS. Les entités autorisées peuvent utiliser et gérer les clés KMS créées par le modèle, soit à l'aide du modèle, de la console AWS KMS ou des API AWS KMS. Lorsque vous supprimez une clé KMS de votre modèle, la suppression de la clé KMS est programmée en utilisant un délai d'attente que vous spécifiez à l'avance.

Par exemple, vous pouvez utiliser un modèle AWS CloudFormation pour créer une clé KMS test avec une politique de clé, une spécification de clé, une utilisation de clé, des alias et des balises de votre choix. Vous pouvez l'exécuter dans votre suite de tests, examiner vos résultats, puis utiliser le modèle pour planifier la suppression de la clé de test. Plus tard, vous pouvez réexécuter le modèle pour créer une clé de test avec les mêmes propriétés.

Vous pouvez également utiliser un modèle AWS CloudFormation pour définir une configuration de clé KMS spécifique qui respecte vos règles métier et vos normes de sécurité. Ensuite, vous pouvez utiliser ce modèle à chaque fois que vous avez besoin de créer une clé KMS. Vous n'avez pas à vous soucier des clés mal configurées. Si votre configuration préférée change, vous pouvez utiliser votre modèle pour mettre à jour vos clés KMS. Par exemple, le modèle facilite l'activation par programmation de la rotation automatique des clés sur toutes les clés KMS définies par le modèle.

Pour obtenir plus d'informations sur les ressources AWS KMS, y compris des exemples, veuillez consulter la référence du type de ressource KMS dans le Guide de l'utilisateur AWS CloudFormation.

En savoir plus sur AWS CloudFormation

Pour en savoir plus sur AWS CloudFormation, consultez les ressources suivantes :

- AWS CloudFormation
- Guide de l’utilisateur AWS CloudFormation
- Référence d'API AWS CloudFormation
- Guide de l’utilisateur de l’interface de ligne de commande AWS CloudFormation

Suppression de AWS KMS keys

La suppression d'une AWS KMS key est destructrice et potentiellement dangereuse. Elle supprime les éléments de clé et toutes les métadonnées associées à la clé KMS, et elle est irréversible. Après la suppression d'une clé KMS, vous ne pouvez plus déchiffrer les données chiffrées sous cette clé, ce qui signifie que les données deviennent irrécupérables. (Les seules exceptions sont les clés de réplica multi-région et les clés asymétriques et KMS HMAC avec un élément de clé importé.) Ce risque est important pour les clés KMS asymétriques utilisées pour le chiffrement où, sans avertissement ni erreur, les utilisateurs peuvent continuer à générer des textes chiffrés avec la clé publique qui ne peuvent pas être déchiffrés une fois la clé privée supprimée de AWS KMS.

Vous devez supprimer une clé KMS seulement lorsque vous êtes sûr de ne plus avoir besoin de l'utiliser. Si vous n'en êtes pas sûr, envisagez de désactiver la clé KMS au lieu de la supprimer. Vous pouvez réactiver une clé KMS désactivée et annuler la suppression planifiée d'une clé KMS, mais vous ne pouvez pas récupérer une clé KMS supprimée.

Vous ne pouvez pas planifier la suppression d'une clé gérée par le client. Vous ne pouvez pas supprimer des Clés gérées par AWS ou Clés détenues par AWS.

Avant de supprimer une clé KMS, vous pouvez découvrir combien de textes chiffrés ont été chiffrés sous cette clé. AWS KMS ne stocke pas ces informations et ne stocke aucun texte chiffré. Pour obtenir ces informations, vous devez déterminer l'utilisation passée d'une clé KMS. Pour obtenir de l'aide, rendez-vous sur Déterminer l'utilisation passée d'une clé KMS.

AWS KMS ne supprime jamais vos clés KMS, sauf si vous les planifiez explicitement pour suppression et que la période d'attente obligatoire expire.

Toutefois, vous pouvez choisir de supprimer une clé KMS pour une ou plusieurs des raisons suivantes :

- Pour terminer le cycle de vie de clés KMS dont vous n'avez plus besoin
- Pour éviter les frais généraux et les coûts de gestion associés à la maintenance des clés KMS inutilisées
- Pour réduire le nombre de clés KMS qui comptent par rapport à votre quota de ressources de clés KMS

> **Note**
>
> Si vous fermez votre Compte AWS, vos clés KMS deviennent inaccessibles et ne vous sont plus facturées.

AWS KMS enregistre une entrée dans votre journal AWS CloudTrail lorsque vous planifiez la suppression de la clé KMS et lorsque la clé KMS est réellement supprimée.

Pour plus d'informations sur la suppression de clés principales et de réplica multi-région, veuillez consulter Suppression de clés multi-régions.

Rubriques

- À propos de la période d'attente
- Suppression des clés KMS asymétriques
- Suppression de clés multi-région
- Suppression des clés KMS avec des éléments de clé importés
- Contrôle de l'accès à une suppression de clé
- Planification et annulation d'une suppression de clé
- Création d'une alarme qui détecte l'utilisation d'une clé KMS en attente de suppression
- Déterminer l'utilisation passée d'une clé KMS

À propos de la période d'attente

Étant donné que la suppression d'une clé KMS est une action destructrice et potentiellement dangereuse, AWS KMS vous oblige à définir un délai d'attente de 7 à 30 jours. La période d'attente par défaut est de 30 jours.

Cependant, la période d'attente réelle peut être jusqu'à 24 heures plus longue celle que vous avez planifiée. Pour obtenir la date et l'heure réelles auxquelles la clé KMS sera supprimée, utilisez l'DescribeKeyopération. Ou, dans la console AWS KMS, dans la page de détails de la clé KMS, dans la section General configuration (Configuration générale), veuillez consulter la Scheduled deletion date (Date de suppression planifiée). Assurez-vous de noter le fuseau horaire.

Pendant la période d'attente, l'état de la clé KMS est Pending deletion (Suppression en attente).

- Une clé KMS qui est en attente de suppression ne peut pas être utilisée dans une opération cryptographique.
- AWS KMS ne soumet pas à une rotation les éléments de clé des clés KMS en attente de suppression.

Après la fin du délai d'attente, AWS KMS supprime la clé KMS, ses alias et toutes les métadonnées AWS KMS associées.

La planification de la suppression d'une clé KMS peut ne pas affecter immédiatement les clés de données chiffrées par la clé KMS. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

Utilisez la période d'attente pour vous assurer de ne pas avoir besoin de la clé KMS maintenant ni par la suite. Vous pouvez configurer une CloudWatch alarme Amazon pour vous avertir si une personne ou une application tente d'utiliser la clé KMS pendant la période d'attente. Pour récupérer la clé KMS, vous pouvez annuler la suppression de clé avant la fin de la période d'attente. Une fois la période d'attente terminée, vous ne pouvez pas annuler la suppression de la clé et AWS KMS supprime la clé KMS.

Suppression des clés KMS asymétriques

Les utilisateurs autorisés peuvent supprimer des clés KMS symétriques ou asymétriques. La procédure pour planifier la suppression de ces clés KMS est la même pour les deux types de clés. Cependant, comme la clé publique d'une clé KMS asymétrique peut être téléchargée et utilisée en dehors d'AWS KMS, l'opération présente des risques supplémentaires importants, en particulier pour les clés KMS asymétriques utilisées pour le chiffrement (l'utilisation de la clé est `ENCRYPT_DECRYPT`).

- Lorsque vous planifiez la suppression d'une clé KMS, l'état de la clé KMS devient Pending deletion (Suppression en attente), et la clé ne peut pas être utilisée dans les opérations cryptographiques. Cependant, la suppression de la planification n'a aucun effet sur les clés publiques en dehors d'AWS KMS. Les utilisateurs disposant de la clé publique peuvent continuer à les utiliser pour chiffrer les messages. Ils ne reçoivent aucune notification indiquant que l'état de la clé est modifié. Sauf si la suppression est annulée, le texte chiffré créé avec la clé publique ne peut pas être déchiffré.
- Les alarmes, les journaux et les autres politiques qui détectent les tentatives d'utilisation de la clé KMS en attente de suppression ne peuvent pas détecter l'utilisation de la clé publique en dehors d'AWS KMS.

- Lorsque la clé KMS est supprimée, toutes les actions AWS KMS impliquant cette clé échouent. Toutefois, les utilisateurs disposant de la clé publique peuvent continuer à les utiliser pour chiffrer les messages. Ces textes chiffrés ne peuvent pas être déchiffrés.

Si vous devez supprimer une clé KMS asymétrique dont la clé d'utilisation est de`ENCRYPT_DECRYPT`, utilisez les entrées de votre CloudTrail journal pour déterminer si la clé publique a été téléchargée et partagée. Si c'est le cas, vérifiez que la clé publique n'est pas utilisée en dehors d'AWS KMS. Ensuite, pensez à désactiver la clé KMS au lieu de la supprimer.

Le risque lié à la suppression d'une clé KMS asymétrique est atténué pour les clés KMS asymétriques avec un élément de clé importé. Pour plus de détails, consultez Suppression d'une clé KMS avec des éléments de clé importés.

Suppression de clés multi-région

Les utilisateurs qui sont autorisés peuvent planifier la suppression des clés principales et de réplica multi-région. Toutefois, AWS KMS ne supprimera pas une clé principale multi-région qui possède des clés de réplica. Aussi, tant que sa clé principale existe, vous pouvez recréer une clé de réplica multi-région supprimée. Pour plus de détails, veuillez consulter Suppression de clés multi-régions.

Suppression des clés KMS avec des éléments de clé importés

Les utilisateurs autorisés peuvent planifier la suppression des clés KMS avec des éléments de clé importés. Cette action supprime définitivement la clé KMS, ses éléments de clé et toutes les métadonnées associées à la clé KMS.

Vous ne pouvez pas créer une nouvelle clé KMS de chiffrement symétrique capable de déchiffrer les textes chiffrés d'une clé de chiffrement symétrique supprimée avec l'élément de clé importé, même si vous disposez d'une copie de son élément de clé. Toutefois, si vous disposez des éléments de clé, vous pouvez recréer de manière efficace une clé KMS asymétrique ou une clé KMS HMAC avec des éléments de clé importés. Pour plus de détails, consultez Suppression d'une clé KMS avec des éléments de clé importés.

Contrôle de l'accès à une suppression de clé

Si vous utilisez des politiques IAM pour accorder des autorisations AWS KMS, toutes les identités IAM qui disposent d'un accès administrateur AWS (`"Action": "*"`) ou d'un accès complet AWS KMS (`"Action": "kms:*"`) sont déjà autorisées à planifier et annuler une suppression de clé pour

les clés KMS. Pour permettre aux administrateurs de clés de planifier et d'annuler la suppression des clés dans la politique de clé, utilisez la console AWS KMS ou l'API AWS KMS.

En général, seuls les administrateurs de clés sont autorisés à planifier ou à annuler la suppression des clés. Vous pouvez toutefois accorder ces autorisations à d'autres identités IAM en ajoutant les autorisations `kms:ScheduleKeyDeletion` et `kms:CancelKeyDeletion` à la politique de clé ou à une politique IAM. Vous pouvez également utiliser la clé de `kms:ScheduleKeyDeletionPendingWindowInDays`condition pour restreindre davantage les valeurs que les principaux peuvent spécifier dans le `PendingWindowInDays` paramètre d'une ScheduleKeyDeletiondemande.

Autoriser les administrateurs de clés à planifier et annuler une suppression de clé (console)

Autoriser les administrateurs de clés à planifier et annuler une suppression de clé.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez l'alias ou l'ID de clé de la clé KMS dont vous voulez modifier les autorisations.
5. Choisissez l'onglet Key policy (Politique de clé).
6. L'étape suivante diffère en ce qui concerne l'affichage par défaut et l'affichage des politiques de votre politique de clé. La vue par défaut n'est disponible que si vous utilisez la politique de clé de console par défaut. Dans le cas contraire, seul l'affichage des politiques est disponible.

   Lorsque la vue par défaut est disponible, un bouton Switch to policy view (Passer à la vue de politique) ou Switch to default view (Passer à la vue par défaut) apparaît dans l'onglet Key policy (Politique de clé).

   - Dans la vue par défaut :

     - Sous Key deletion (Suppression de clé), sélectionnez Allow key administrators to delete this key (Autoriser les administrateurs de clé à supprimer cette clé).
   - Dans la vue de politique :

     a. Choisissez Edit (Modifier).

b. Dans l'instruction de politique destinée aux administrateurs de clés, ajoutez les autorisations `kms:ScheduleKeyDeletion` et `kms:CancelKeyDeletion` à l'élément `Action`.

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

c. Sélectionnez Enregistrer les modifications.

Autoriser les administrateurs de clés à planifier et à annuler la suppression de clés (AWS CLI)

Vous pouvez utiliser AWS Command Line Interface pour ajouter des autorisations pour la planification et l'annulation d'une suppression de clé.

Pour ajouter une autorisation pour planifier et annuler une suppression de clé

1. Utilisez la commande `aws kms get-key-policy` pour récupérer la politique de clé existante, puis enregistrez le document de politique dans un fichier.
2. Ouvrez le document de stratégie dans votre éditeur de texte préféré. Dans l'instruction de politique destinée aux administrateurs de clés, ajoutez les autorisations

`kms:ScheduleKeyDeletion` et `kms:CancelKeyDeletion`. L'exemple suivant montre une déclaration de politique avec les deux autorisations suivantes :

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

3. Utilisez la commande `aws kms put-key-policy` pour appliquer la politique de clé à la clé KMS.

Planification et annulation d'une suppression de clé

Les procédures suivantes décrivent comment planifier une suppression de clé et annuler la suppression de AWS KMS keys (clés KMS) à région unique dans AWS KMS à l'aide de la AWS Management Console, de l'AWS CLI et de AWS SDK for Java.

Pour plus d'informations sur la planification de la suppression de clés multi-région, veuillez consulter Suppression de clés multi-régions.

> **Warning**
>
> La suppression d'une clé KMS est destructrice et potentiellement dangereuse. Vous devez y avoir recours seulement lorsque vous êtes sûr de ne plus avoir besoin d'utiliser la clé KMS

maintenant ni par la suite. Si vous n'en êtes pas sûr, vous devriez désactiver la clé KMS au lieu de la supprimer.

Avant de pouvoir supprimer une clé KMS, vous devez avoir la permission de le faire. Pour plus d'informations sur l'octroi de ces autorisations aux administrateurs de clés, veuillez consulter la rubrique Contrôle de l'accès à une suppression de clé. Vous pouvez également utiliser la clé de condition `kms:ScheduleKeyDeletionPendingWindowInDays` pour limiter davantage le délai d'attente, par exemple en imposant un délai d'attente minimum.

AWS KMS enregistre une entrée dans votre journal AWS CloudTrail lorsque vous planifiez la suppression de la clé KMS et lorsque la clé KMS est réellement supprimée.

Planification et annulation d'une suppression de clé (console)

Dans la AWS Management Console, vous pouvez planifier et annuler la suppression de plusieurs clés KMS à la fois.

Pour planifier une suppression de clé

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.

   Vous ne pouvez pas planifier la suppression deClés gérées par AWS ou de Clés détenues par AWS.
4. Cochez la case en regard de la clé KMS que vous souhaitez supprimer.
5. Choisissez Actions de clé, Planifier une suppression de clé.
6. Lisez et tenez compte de l'avertissement et des informations sur l'annulation de la suppression pendant la période d'attente. Si vous décidez d'annuler la suppression, en bas de la page, sélectionnez Cancel (Annuler).
7. Pour Période d'attente (en jours), tapez un nombre de jours compris entre 7 et 30.
8. Vérifiez les clés KMS que vous supprimez.
9. Cochez la case en regard de Confirm you want to schedule this key for deletion in ***<number of days>*** days. (Confirmez que cette clé doit être supprimée dans <number of days> jours).

10. Choisissez Schedule deletion (Planifier la suppression).

L'état de la clé KMS passe à Pending deletion (En attente de suppression).

Pour annuler une suppression de clé

1. Ouvrez la console AWS KMS à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Cochez la case en regard de la clé KMS que vous souhaitez récupérer.
5. Choisissez Actions de clé, Annuler la suppression de clé.

L'état de la clé KMS passe de Pending deletion (En attente de suppression) à Disabled (Désactivé). Pour utiliser la clé KMS, vous devez l'activer.

Planification et annulation d'une suppression de clé (AWS CLI)

Utilisez la commande `aws kms schedule-key-deletion` pour planifier une suppression de clé gérée par le client, comme illustré dans l'exemple suivant.

Vous ne pouvez pas planifier la suppression d'une Clé gérée par AWS ou d'une Clé détenue par AWS.

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --
pending-window-in-days 10
```

Lorsqu'elle est utilisée avec succès, l'AWS CLI renvoie une sortie similaire à celle affichée dans l'exemple suivant :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": 1598304792.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 10
}
```

Utilisez la commande `aws kms cancel-key-deletion` pour annuler une suppression de clé à partir de l'AWS CLI, comme illustré dans l'exemple suivant.

```
$ aws kms cancel-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Lorsqu'elle est utilisée avec succès, l'AWS CLI renvoie une sortie similaire à celle affichée dans l'exemple suivant :

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

L'état de la clé KMS passe de Pending Deletion (En attente de suppression) à Disabled (Désactivé). Pour utiliser la clé KMS, vous devez l'activer.

Planification et annulation d'une suppression de clé (AWS SDK for Java)

L'exemple suivant montre comment planifier la suppression d'une clé gérée par le client à l'aide de AWS SDK for Java. Cet exemple exige que vous instanciiez auparavant un client `AWSKMSClient` en tant que `kms`.

```
String KeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

int PendingWindowInDays = 10;

ScheduleKeyDeletionRequest scheduleKeyDeletionRequest =
new
 ScheduleKeyDeletionRequest().withKeyId(KeyId).withPendingWindowInDays(PendingWindowInDays);
kms.scheduleKeyDeletion(scheduleKeyDeletionRequest);
```

L'exemple suivant montre comment annuler la suppression d'une clé à l'aide du kit AWS SDK for Java. Cet exemple exige que vous instanciiez auparavant un client `AWSKMSClient` en tant que `kms`.

```
String KeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CancelKeyDeletionRequest cancelKeyDeletionRequest =
new CancelKeyDeletionRequest().withKeyId(KeyId);
```

```
kms.cancelKeyDeletion(cancelKeyDeletionRequest);
```

L'état de la clé KMS passe de Pending Deletion (En attente de suppression) à Disabled (Désactivé). Pour utiliser la clé KMS, vous devez l'activer.

Création d'une alarme qui détecte l'utilisation d'une clé KMS en attente de suppression

Vous pouvez combiner les fonctionnalités d'AWS CloudTrailAmazon CloudWatch Logs et d'Amazon Simple Notification Service (Amazon SNS) pour créer une alarme CloudWatch Amazon qui vous avertit lorsqu'un utilisateur de votre compte essaie d'utiliser une clé KMS en attente de suppression. Si vous recevez cette notification, vous pouvez annuler la suppression de la clé KMS et reconsidérer votre décision de la supprimer.

Les procédures suivantes créent une alarme qui vous avertit chaque fois que le message d'erreur *`Key ARN`* `is pending deletion` « » est écrit dans vos fichiers CloudTrail journaux. Ce message d'erreur indique qu'une personne ou une application tente d'utiliser la clé KMS dans une opération de chiffrement. Étant donné que la notification est liée au message d'erreur, elle n'est pas déclenchée lorsque vous utilisez des opérations d'API autorisées sur les clés KMS en attente de suppression, telles que `ListKeys`, `CancelKeyDeletion` et `PutKeyPolicy`. Pour afficher la liste des opérations d'API AWS KMS qui retournent ce message d'erreur, consultez Principaux états des clés AWS KMS.

L'e-mail de notification que vous recevez ne répertorie pas la clé KMS ou les opérations de chiffrement. Vous pouvez trouver ces informations dans votre journal CloudTrail. Au lieu de cela, l'e-mail indique que l'état de l'alarme est passé de OK à Alarme. Pour plus d'informations sur les CloudWatch alarmes et les changements d'état, consultez la section Utilisation des CloudWatch alarmes Amazon dans le guide de CloudWatch l'utilisateur Amazon.

> **Warning**
>
> Cette CloudWatch alarme Amazon ne peut pas détecter l'utilisation de la clé publique d'une clé KMS asymétrique en dehors deAWS KMS. Pour plus de détails sur les risques particuliers liés à la suppression de clés KMS asymétriques utilisées pour le chiffrement de la clé publique, en particulier la création de textes chiffrés qui ne peuvent pas être déchiffrés, veuillez consulter Suppression des clés KMS asymétriques.

Rubriques

- Exigences relatives à une CloudWatch alarme
- Création de l' CloudWatch alarme

Exigences relatives à une CloudWatch alarme

Avant de créer une CloudWatch alarme, vous devez créer un journal et AWS CloudTrail le configurer CloudTrail pour envoyer les fichiers CloudTrail CloudWatch journaux à Amazon Logs. Vous avez également besoin d'une rubrique Amazon SNS pour la notification d'alarme.

- Créer un suivi CloudTrail

  CloudTrail est automatiquement activé sur votre compte Compte AWS lorsque vous créez le compte. Toutefois, pour un enregistrement continu des événements dans votre compte, y compris les événements pour AWS KMS, créez un journal de suivi.
- Configurez CloudTrail pour fournir vos fichiers CloudWatch journaux Logs.

  Configurez la livraison de vos fichiers CloudTrail CloudWatch journaux à Logs. Cela permet à CloudWatch Logs de surveiller les journaux pour AWS KMS détecter les demandes d'API qui tentent d'utiliser une clé KMS en attente de suppression.
- Créer une rubrique Amazon SNS.

  Lorsque votre alarme se déclenche, elle vous avertit en envoyant un message à une adresse e-mail figurant dans une rubrique Amazon Simple Notification Service (Amazon SNS).

Création de l' CloudWatch alarme

Dans cette procédure, vous créez un filtre métrique de groupe de CloudWatch journaux qui recherche les instances de l'exception de suppression en attente. Ensuite, vous créez une CloudWatch alarme en fonction de la métrique du groupe de logs. Pour plus d'informations sur les filtres métriques des groupes de journaux, consultez la section Création de métriques à partir d'événements de journal à l'aide de filtres dans le guide de l'utilisateur Amazon CloudWatch Logs.

1. Créez un filtre CloudWatch métrique qui analyse les CloudTrail journaux.

   Suivez les instructions de la section Créer un filtre métrique pour un groupe de journaux à l'aide des valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Modèle de filtre	`{ $.eventSource = kms* && $.errorMessage = "* is pending deletion."}`
Valeur de la métrique	1

2. Créez une CloudWatch alarme en fonction du filtre métrique que vous avez créé à l'étape 1.

   Suivez les instructions de la section Création d'une CloudWatch alarme basée sur un filtre métrique de groupes de logs en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Filtre de métrique	Le nom du filtre de métrique que vous avez créé à l'étape 1.
Type de seuil	Statique
Conditions	Chaque fois que le *`nom de la métrique`* est supérieur à 1
Points de données à alarmer	1 sur 1
Traitement de données manquantes	Traiter les données manquantes comme correctes (seuil non dépassé)

Une fois cette procédure terminée, vous recevrez une notification chaque fois que votre nouvelle CloudWatch alarme entre dans l'`ALARM`état. Si vous recevez une notification pour cette alarme, cela peut signifier qu'une clé KMS dont la suppression est planifiée est toujours nécessaire pour chiffrer ou déchiffrer les données. Dans ce cas, annulez la suppression de la clé KMS et reconsidérez votre décision de la supprimer.

Déterminer l'utilisation passée d'une clé KMS

Avant de supprimer une clé KMS, vous pouvez découvrir combien de textes chiffrés ont été chiffrés sous cette clé. AWS KMS ne stocke pas ces informations et ne stocke aucun texte chiffré. Savoir comment une clé KMS a été utilisée dans le passé peut vous aider à décider si vous en aurez besoin ou non à l'avenir. Cette rubrique propose plusieurs politiques qui peuvent vous aider à déterminer l'utilisation passée d'une clé KMS.

> **Warning**
>
> Ces politiques pour déterminer l'utilisation passée et actuelle ne sont efficaces que pour les utilisateurs AWS et les opérations AWS KMS. Elles ne peuvent pas détecter l'utilisation de la clé publique d'une clé KMS asymétrique en dehors d'AWS KMS. Pour plus de détails sur les risques particuliers liés à la suppression de clés KMS asymétriques utilisées pour le chiffrement de la clé publique, en particulier la création de textes chiffrés qui ne peuvent pas être déchiffrés, veuillez consulter Suppression des clés KMS asymétriques.

Rubriques

- Examen des autorisations d'une clé KMS afin de déterminer la portée potentielle de son utilisation
- Examen des journaux AWS CloudTrail pour déterminer l'utilisation réelle

Examen des autorisations d'une clé KMS afin de déterminer la portée potentielle de son utilisation

Déterminer qui a actuellement accès à une clé KMS peut vous aider à déterminer l'ampleur de l'utilisation passée de cette clé KMS et si elle est encore requise. Pour découvrir comment déterminer qui a actuellement accès à une clé KMS, consultez la rubrique Déterminer l'accès à des AWS KMS keys.

Examen des journaux AWS CloudTrail pour déterminer l'utilisation réelle

Vous pouvez éventuellement utiliser un historique d'utilisation de clé KMS pour déterminer si vous disposez de textes chiffrés sous une clé KMS particulière.

L'ensemble des activités d'API AWS KMS est enregistré dans des fichiers journaux AWS CloudTrail. Si vous avez créé un suivi CloudTrail dans la région où se trouve votre clé KMS, vous pouvez examiner vos fichiers CloudTrail journaux pour consulter l'historique de toutes les activités d'AWS

KMSAPI relatives à une clé KMS en particulier. Si vous n'avez pas de parcours, vous pouvez toujours consulter les événements récents dans l'historique de vos CloudTrail événements. Pour plus de détails sur la façon dont AWS KMS les utilisations sont CloudTrail utilisées, voirJournalisation des appels d'API AWS KMS avec AWS CloudTrail.

Les exemples suivants montrent les entrées de CloudTrail journal générées lorsqu'une clé KMS est utilisée pour protéger un objet stocké dans Amazon Simple Storage Service (Amazon S3). Dans cet exemple, l'objet est chargé vers Simple Storage Service (Amazon S3) au moyen de la Protection des données à l'aide du chiffrement côté serveur avec des clés KMS (SSE-KMS). Lorsque vous chargez un objet sur Amazon S3 avec SSE-KMS, vous spécifiez la clé KMS à utiliser pour protéger l'objet. Amazon S3 utilise l'AWS KMSGenerateDataKeyopération pour demander une clé de données unique pour l'objet, et cet événement de demande est enregistré CloudTrail avec une entrée similaire à la suivante :


```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:18Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
```

```
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"},
    "keySpec": "AES_256",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "cea04450-5817-11e5-85aa-97ce46071236",
  "eventID": "80721262-21a5-49b9-8b63-28740e7ce9c9",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Lorsque vous téléchargez ultérieurement cet objet à partir d'Amazon S3, Amazon S3 envoie une demande `Decrypt` à AWS KMS pour déchiffrer la clé de données de l'objet à l'aide de la clé KMS spécifiée. Dans ce cas, vos fichiers CloudTrail journaux incluent une entrée similaire à la suivante :

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
```

```
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:39Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"}},
  "responseElements": null,
  "requestID": "db750745-5817-11e5-93a6-5b87e27d91a0",
  "eventID": "ae551b19-8a09-4cfc-a249-205ddba330e3",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Toute l'activité d'API AWS KMS est consignée par CloudTrail. En évaluant ces entrées de journal, vous pouvez éventuellement déterminer l'utilisation passée d'une clé KMS particulière et cela peut vous aider à déterminer si vous souhaitez la supprimer ou non.

Pour voir d'autres exemples illustrant la façon dont l'activité des AWS KMS API apparaît dans vos fichiers CloudTrail journaux, rendez-vous surJournalisation des appels d'API AWS KMS avec AWS CloudTrail. Pour plus d'informations à ce sujet, CloudTrail consultez le guide de AWS CloudTrail l'utilisateur.

Principaux états des clés AWS KMS

Une AWS KMS key a toujours un état de clé. Les opérations sur la clé KMS et son environnement peuvent modifier cet état de clé, soit de manière transitoire, soit jusqu'à ce qu'une autre opération modifie son état de clé.

Le tableau de cette section montre comment les états de clé affectent les appels vers les opérations d'API AWS KMS. En raison de son état clé, une opération sur une clé KMS devrait réussir (#),

échouer (X), ou ne réussir que dans certaines conditions (?). Le résultat est souvent différent pour les clés KMS avec des éléments de clé importés.

Ce tableau inclut uniquement les opérations d'API qui utilisent une clé KMS existante. Les autres opérations, telles que CreateKeyet ListKeys, sont omises.

Rubriques

- États de clé et types de clés KMS
- Tableau d'état de clé

États de clé et types de clés KMS

Le type de la clé KMS détermine les états de clé qu'elle peut avoir.

- Toutes les clés KMS peuvent être à l'état `Enabled`, `Disabled` et `PendingDeletion`.
- La plupart des clés KMS sont créées dans l'état `Enabled`. Les clés avec des éléments de clé importés sont créées dans l'état `PendingImport`.
- L'état `PendingImport` s'applique uniquement aux clés KMS avec des éléments de clé importés.
- L'état `Unavailable` s'applique uniquement à une clé KMS dans un magasin de clés personnalisé. Une clé KMS dans un magasin de clés AWS CloudHSM est `Unavailable` lorsque le magasin de clés personnalisé est intentionnellement déconnecté de son cluster AWS CloudHSM. Une clé KMS dans un magasin de clés externe est `Unavailable` lorsque le magasin de clés personnalisé est intentionnellement déconnecté de son proxy de magasin de clés externe. Vous pouvez afficher et gérer les clés KMS indisponibles, mais vous ne pouvez pas les utiliser dans les opérations de chiffrement.

  L'état d'une clé KMS dans un magasin de clés personnalisé n'est pas affecté par les modifications apportées à sa clé de sauvegarde. Une clé KMS dans un magasin de clés AWS CloudHSM n'est pas affectée par les modifications apportées à ses éléments de clé associés dans le cluster AWS CloudHSM. Une clé KMS dans un magasin de clés externe n'est pas affectée par les modifications apportées à sa clé externe dans un gestionnaire de clés externe. Si la clé de sauvegarde est désactivée ou supprimée, l'état de la clé KMS ne change pas, mais les opérations cryptographiques utilisant la clé KMS échouent.
- Les états de clé `Creating`, `Updating` et `PendingReplicaDeletion` s'appliquent uniquement aux clés multi-région.

- Une clé de réplica multi-région se trouve dans l'état de clé `Creating` durant sa création. Ce processus est peut-être toujours en cours une fois l'ReplicateKeyopération terminée. Lorsque le processus de réplication est terminé, la clé de réplica se trouve dans l'état `Enabled` ou `PendingImport`.
- Les clés multi-région se trouvent à l'état transitoire `Updating` lorsque la région principale est en cours de mise à jour. Ce processus est peut-être toujours en cours une fois l'UpdatePrimaryRegionopération terminée. Une fois le processus de mise à jour terminé, les clés principales et de réplica reprennent l'état de clé `Enabled`.
- Lorsque vous planifiez la suppression d'une clé principale multi-région dotée de clés de réplica, la clé principale se trouve à l'état `PendingReplicaDeletion` jusqu'à ce que toutes ses clés de réplica soient supprimées. Puis, son état passe à `PendingDeletion`. Pour plus de détails, veuillez consulter Suppression de clés multi-régions.

Tableau d'état de clé

Le tableau suivant montre comment l'état de clé d'une clé KMS affecte les opérations AWS KMS.

Les descriptions des notes de bas de page numérotées ([n]) sont à la fin de cette rubrique.

> **Note**
>
> Vous devrez peut-être faire défiler horizontalement ou verticalement pour voir toutes les données de ce tableau.

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
CancelKeyDeletion							

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
	[4]	[4]		[4]	[4], [13]	[4]	[4]
CreateAlias	✓	✓	⊘ [3]	✓	✓	✓	✓
CreateGrant	✓	⊘ [1]	⊘ [2] ou [3]	⊘ [5]	✓	⊘ [14]	✓
Decrypt	✓	⊘ [1]	⊘ [2] ou [3]	⊘ [5]	⊘ [11]	⊘ [14]	✓
DeleteAlias	✓	✓	✓	✓	✓	✓	✓
DeleteImportedKeyMaterial	✓ [9]	✓ [9]	✓ [9]	✓ (Aucun effet)	N/A	⊘ [14]	⊘ [15]
DescribeKey	✓	✓	✓	✓	✓	✓	✓

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
DisableKey	✓	✓	⊗ [3]	⊗ [5]	✓ [12]	⊗ [14]	⊗ [15]
DisableKeyRotation	? [7]	⊗ [1] ou [7]	⊗ [3] ou [7]	⊗ [6]	⊗ [7]	⊗ [14]	? [7]
EnableKey	✓	✓	⊗ [3]	⊗ [5]	✓ [12]	⊗ [14]	⊗ [15]
EnableKeyRotation	? [7]	⊗ [1] ou [7]	⊗ [3] ou [7]	⊗ [6]	⊗ [7]	⊗ [14]	? [7]
Encrypt	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKey	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
GenerateDataKeyPair	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateMac	✓	⊗ [1]	⊗ [2] ou [3]	N/A	N/A	⊗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	? [7]	? [7]	? [7]	⊗ [6]	⊗ [7]	? [7]	? [7]

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
GetParametersForImport	[9]	[9]	[8] ou [9]		[9]	[14]	[15]
GetPublicKey		[1]	[2] ou [3]	N/A	N/A	[14]	
ImportKeyMaterial	[9]	[9]	[8] ou [9]		[9]	[14]	
ListAliases							
ListGrants							
ListKeyPolicies							
ListResourceTags							
PutKeyPolicy							

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
ReEncrypt	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
Replicate Key	✓	⊗ [1]	⊗ [2] ou [3]	⊗ [5]	N/A	⊗ [14]	⊗ [15]
RetireGrant	✓	✓	✓	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
ScheduleKeyDeletion	✓	✓	⊗ [3]	✓	✓	✓	⊗ [15]
Sign (Signer)	✓	⊗ [1]	⊗ [2] ou [3]	N/A	N/A	⊗ [14]	✓
TagResource	✓	✓	⊗ [3]	✓	✓	✓	✓

API	Activé	Désactivées	Suppression en attente Suppression du réplica en attente	Importation en attente	Unavailable	Création	Mise à jour
UntagResource	✓	✓	⊘ [3]	✓	✓	✓	✓
UpdateAlias	✓	✓	? [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	⊘ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊘ [1]	⊘ [2] ou [3]	⊘ [5]	N/A	⊘ [14]	✓
Vérification	✓	⊘ [1]	⊘ [2] ou [3]	N/A	N/A	⊘ [14]	✓
VerifyMac	✓	⊘ [1]	⊘ [2] ou [3]	N/A	N/A	⊘ [14]	✓

Détails de la table

- [1] `DisabledException: <key ARN> is disabled.`
- [2] `DisabledException: <key ARN> is pending deletion (or pending replica deletion).`
- [3] `KMSInvalidStateException: <key ARN> is pending deletion (or pending replica deletion).`
- [4] `KMSInvalidStateException: <key ARN> is not pending deletion (or pending replica deletion).`
- [5] `KMSInvalidStateException: <key ARN> is pending import.`
- [6] `UnsupportedOperationException: <key ARN> origin is EXTERNAL which is not valid for this operation.`
- [7] Si la clé KMS possède des éléments de clé importés ou se trouve dans un magasin de clés personnalisé : `UnsupportedOperationException`.
- [8] Si la clé KMS comporte des éléments de clé importés : `KMSInvalidStateException`
- [9] Si la clé KMS ne peut pas comporter ou ne comporte pas des éléments de clé importés : `UnsupportedOperationException`.
- [10] Si la clé KMS source est en attente de suppression, la commande réussit. Si la clé KMS de destination est en attente de suppression, la commande échoue avec l'erreur suivante : `KMSInvalidStateException : <key ARN> is pending deletion.`
- [11] `KMSInvalidStateException: <key ARN> is unavailable.` Vous ne pouvez pas effectuer cette opération sur une clé KMS indisponible.
- [12] L'opération aboutit, mais l'état de la clé KMS ne change pas jusqu'à ce qu'elle devienne disponible.
- [13] Même si une clé KMS d'un magasin de clés personnalisé est en attente de suppression, son état de clé demeure `PendingDeletion`, même si la clé KMS devient indisponible. Cela vous permet d'annuler la suppression de la clé KMS à tout moment au cours de la période d'attente.
- [14] `KMSInvalidStateException: <key ARN> is creating.` AWS KMS lève cette exception alors qu'il réplique une clé multi-région (`ReplicateKey`).
- [15] `KMSInvalidStateException: <key ARN> is updating.` AWS KMS lève cette exception pendant qu'il met à jour la région principale d'une clé multi-région (`UpdatePrimaryRegion`).

Authentification et contrôle d'accès pour AWS KMS

Pour utiliser AWS KMS, vous devez posséder des informations d'identification que AWS peut utiliser pour authentifier vos demandes. Les informations d'identification doivent inclure des autorisations pour accéder aux ressources AWS, telles que les AWS KMS keys et alias. Aucun principal AWS n'est autorisé sur une clé KMS, sauf si cette autorisation est fournie explicitement et jamais refusée. Il n'existe aucune autorisation implicite ou automatique pour utiliser ou gérer une clé KMS.

Le principal moyen de gérer l'accès à vos ressources AWS KMS consiste à utiliser des politiques. Les politiques sont des documents qui décrivent quels principaux peuvent accéder à quelles ressources. Les politiques attachées à une identité IAM sont appelées politiques basées sur l'identité (ou politiques IAM), et celles attachées à d'autres types de ressources sont appelées politiques de ressources. Les politiques de ressources AWS KMS pour les clés KMS sont appelées politiques de clé. Toutes les clés KMS ont une politique de clé.

Pour contrôler l'accès à vos alias AWS KMS, utilisez des politiques IAM. Pour autoriser les principaux à créer des alias, vous devez fournir l'autorisation d'accéder à l'alias dans une politique IAM et l'autorisation d'accéder à la clé dans une politique de clé. Pour plus de détails, consultez Contrôle de l'accès aux alias.

Pour contrôler l'accès à vos clés KMS, vous pouvez utiliser les mécanismes de politique suivants.

- Politique de clé – chaque clé KMS a une politique de clé. Il s'agit du principal mécanisme de contrôle d'accès à une clé KMS. Vous pouvez utiliser la politique de clé seule pour contrôler l'accès, ce qui signifie que l'étendue complète de l'accès à la clé KMS est définie dans un document unique (la politique de clé). Pour plus d'informations sur l'utilisation des politiques de clé, consultez la rubrique Politiques de clé.
- Politiques IAM – vous pouvez utiliser des politiques IAM en combinaison avec la politique de clé et les octrois pour contrôler l'accès à une clé KMS. Contrôler l'accès de cette façon vous permet de gérer toutes les autorisations pour vos identités IAM dans IAM. Pour utiliser une politique IAM pour autoriser l'accès à une clé KMS, la politique de clé doit l'autoriser explicitement. Pour en savoir plus sur l'utilisation de politiques IAM, veuillez consulter Politiques IAM.
- Octrois – Vous pouvez utiliser des octrois en association avec les politiques IAM et de clé pour autoriser l'accès à une clé KMS. En contrôlant l'accès de cette façon, vous pouvez autoriser l'accès à la clé KMS dans la politique de clé et permettre aux identités de déléguer leur accès à d'autres personnes. Pour plus d'informations sur l'utilisation des octrois, consultez la rubrique Octrois dans AWS KMS.

Les clés KMS appartiennent au compte AWS dans lequel elles ont été créées. Cependant, aucune identité ni aucun principal, y compris l'utilisateur root du compte AWS, n'est autorisé à utiliser ou à gérer une clé KMS, sauf si cette autorisation est explicitement fournie dans une politique de clé, une politique IAM ou une autorisation. L'identité IAM qui crée une clé KMS n'est pas considérée comme le propriétaire de la clé et elle n'est pas automatiquement autorisée à utiliser ou à gérer la clé KMS qu'elle a créée. Comme toute autre identité, le créateur de la clé doit obtenir une autorisation par le biais d'une politique de clé, d'une politique IAM ou d'un octroi. Toutefois, les identités qui disposent de l'autorisation `kms:CreateKey` peuvent définir la politique de clé initiale et s'octroyer l'autorisation d'utiliser ou de gérer la clé.

Les rubriques suivantes fournissent des détails sur la façon dont vous pouvez utiliser les autorisations AWS Identity and Access Management (IAM) et AWS KMS pour sécuriser vos ressources en contrôlant qui peut y accéder.

Rubriques

- Concepts dans le contrôle d'accès AWS KMS
- Politiques de clé dans AWS KMS
- Utilisation des politiques IAM avec AWS KMS
- Octrois dans AWS KMS
- Connexion à AWS KMS via un point de terminaison d'un VPC
- Clés de condition pour AWS KMS
- ABAC pour AWS KMS
- Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS
- Utilisation des rôles liés aux services pour AWS KMS
- Utilisation du TLS post-quantique hybride avec AWS KMS
- Déterminer l'accès à des AWS KMS keys
- AWS KMS autorisations
- Test des autorisations

Concepts dans le contrôle d'accès AWS KMS

Découvrez les concepts utilisés dans les discussions sur le contrôle d'accès dans AWS KMS.

Rubriques

- Authentification
- Autorisation
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Ressources AWS KMS

Authentification

L'authentification est le processus de vérification de votre identité. Pour envoyer une demande à AWS KMS, vous devez vous connecter à AWS à l'aide de vos informations d'identification AWS.

Autorisation

L'autorisation permet d'envoyer des demandes de création, de gestion ou d'utilisation des ressources AWS KMS. Par exemple, vous devez être autorisé à utiliser une clé KMS dans le cadre d'une opération de chiffrement.

Pour contrôler l'accès à vos ressources AWS KMS, utilisez des politiques de clé, des politiques IAM et des octrois. Chaque clé KMS doit avoir une politique de clé. Si la politique de clé le permet, vous pouvez également utiliser des octrois et des politiques IAM pour accorder aux principaux l'accès à la clé KMS. Pour affiner votre autorisation, vous pouvez utiliser des clés de condition qui autorisent ou refusent l'accès uniquement lorsqu'une demande ou une ressource remplit les conditions que vous définissez. Vous pouvez également autoriser l'accès aux principaux auxquels vous faites confiance sur d'autres Comptes AWS.

Authentification par des identités

L’authentification correspond au processus par lequel vous vous connectez à AWS avec vos informations d’identification. Vous devez vous authentifier (être connecté à AWS) en tant qu’Utilisateur racine d'un compte AWS, en tant qu’utilisateur IAM ou en endossant un rôle IAM.

Vous pouvez vous connecter à AWS en tant qu’identité fédérée à l’aide des informations d’identification fournies par le biais d’une source d’identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l’authentification de connexion unique de votre entreprise et vos informations d’identification Google ou Facebook sont des exemples d’identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d’identités avec des rôles IAM. Lorsque vous accédez à AWS en utilisant la fédération, vous endossez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter à la AWS Management Console ou au portail d'accès AWS. Pour plus d'informations sur la connexion à AWS, consultez Connexion à votre Compte AWSdans le Guide de l'utilisateurConnexion à AWS.

Si vous accédez à AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes en utilisant vos informations d'identification. Si vous n'utilisez pas les outils AWS, vous devez signer les requêtes vous-même. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez Signature des demandes d'API AWS dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multifactorielle (MFA) pour améliorer la sécurité de votre compte. Pour en savoir plus, veuillez consulter Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Utilisateur root Compte AWS

Lorsque vous créez un Compte AWS, vous commencez avec une seule identité de connexion disposant d'un accès complet à tous les Services AWS et ressources du compte. Cette identité est appelée utilisateur root du Compte AWS. Vous pouvez y accéder en vous connectant à l'aide de l'adresse électronique et du mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur root pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur root et utilisez-les pour effectuer les tâches que seul l'utilisateur root peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant des informations d'identification d'utilisateur root dans le Guide de l'utilisateur IAM.

Identité fédérée

Demandez aux utilisateurs humains, et notamment aux utilisateurs qui nécessitent un accès administrateur, d'appliquer la bonne pratique consistant à utiliser une fédération avec fournisseur d'identité pour accéder à Services AWS en utilisant des informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, un fournisseur d'identité Web, l'AWS Directory Service, l'annuaire Identity Center ou tout utilisateur qui accède à Services AWS en utilisant des informations d'identification fournies via une source d'identité. Quand

des identités fédérées accèdent à Comptes AWS, elles endossent des rôles, ces derniers fournissant des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous connecter et vous synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité pour une utilisation sur l'ensemble de vos applications et de vos Comptes AWS. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center.

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité dans votre Compte AWS qui dispose d'autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une entité au sein de votre Compte AWS qui dispose d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez temporairement endosser un rôle IAM dans la AWS Management Console en changeant de rôle. Vous pouvez obtenir un rôle en appelant une opération d'API AWS CLI ou

AWS à l'aide d'une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center.
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, certains Services AWS vous permettent d'attacher une politique directement à une ressource (au lieu d'utiliser un rôle en tant que proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès interservices : certains Services AWS utilisent des fonctions dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, une fonction du service ou un rôle lié au service.
  - Forward access sessions (FAS) – Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions dans AWS, vous êtes considéré comme un principal. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche une autre action dans un autre service. FAS utilise les autorisations du principal appelant Service AWS, combinées à la demande Service AWS pour effectuer des demandes aux services en aval. Les demandes FAS ne sont formulées que lorsqu'un service reçoit une demande qui, pour aboutir, a besoin d'interagir avec d'autres ressources ou Services AWS. Dans ce cas, vous devez disposer

d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.

- Fonction du service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
- Rôle lié au service – Un rôle lié au service est un type de fonction du service lié à un Service AWS. Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre Compte AWS et sont détenus par le service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications s'exécutant sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer des informations d'identification temporaires pour les applications s'exécutant sur une instance EC2 et effectuant des demandes d'API AWS CLI ou AWS. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un rôle AWS à une instance EC2 et le rendre disponible à toutes les applications associées, vous pouvez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez les accès dans AWS en créant des politiques et en les attachant à des identités AWS ou à des ressources. Une politique est un objet dans AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit les autorisations de ces dernières. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur racine ou séance de rôle) envoie une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées dans AWS en tant que documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Présentation des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur avec cette politique peut obtenir des informations utilisateur à partir de la AWS Management Console, de la AWS CLI ou de l'API AWS.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez attacher à plusieurs utilisateurs, groupes et rôles dans votre Compte AWS. Les politiques gérées incluent les politiques gérées par AWS et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Une politique de clé AWS KMS est une politique basée sur les ressources qui contrôle l'accès à une clé KMS. Chaque clé KMS doit avoir une politique de clé. Vous pouvez utiliser un autre mécanisme d'autorisation pour autoriser l'accès à la clé KMS, mais uniquement si la politique de clé le permet. (Vous pouvez utiliser une politique IAM pour refuser l'accès à une clé KMS, même si la politique de clé ne l'autorise pas explicitement.)

Les politiques basées sur les ressources sont des documents de politique JSON que vous joignez à une ressource, telle qu'une clé KMS, pour contrôler l'accès à cette ressource. La politique basée

sur les ressources définit les actions qu'un principal donné peut effectuer sur cette ressource et dans quelles conditions. La ressource ne peut pas être spécifiée dans une politique basée sur les ressources. En revanche, vous devez y spécifier un principal (comptes, utilisateurs, rôles, utilisateurs fédérés ou Services AWS). Les politiques basées sur les ressources sont des politiques en ligne qui sont situées dans le service qui gère la ressource. Vous ne pouvez pas utiliser les politiques gérées par AWS depuis IAM, comme la politique gérée par `AWSKeyManagementServicePowerUser` dans une politique basée sur les ressources.

Listes de contrôle d’accès (ACL)

Les listes de contrôle d’accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l’autorisation d’accéder à une ressource. Les listes de contrôle d’accès sont similaires aux politiques basées sur les ressources, bien qu’elles n’utilisent pas le format de document de politique JSON.

Amazon S3, AWS WAF et Amazon VPC sont des exemples de services prenant en charge les ACL. Pour en savoir plus sur les listes de contrôle d’accès, consultez Présentation des listes de contrôle d’accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

AWS KMS ne prend pas en charge les listes de contrôle d'accès.

Autres types de politique

AWS prend en charge d’autres types de politiques moins courantes. Ces types de politiques peuvent définir le nombre maximum d’autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d’autorisations : une limite d’autorisations est une fonction avancée dans laquelle vous définissez le nombre maximal d’autorisations qu’une politique basée sur l’identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d’autorisations pour une entité. Les autorisations qui en résultent représentent la combinaison des politiques basées sur l’identité d’une entité et de ses limites d’autorisation. Les politiques basées sur les ressources qui spécifient l’utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d’autorisations. Un refus explicite dans l’une de ces politiques remplace l’autorisation. Pour plus d’informations sur les limites d’autorisations, consultez Limites d’autorisations pour des entités IAM dans le Guide de l’utilisateur IAM.
- Politiques de contrôle des services (SCP) - les SCP sont des politiques JSON qui spécifient le nombre maximal d’autorisations pour une organisation ou une unité d’organisation (OU) dans AWS Organizations. AWS Organizations est un service qui vous permet de regrouper et de gérer de

façon centralisée plusieurs Comptes AWS détenus par votre entreprise. Si vous activez toutes les fonctions d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. La SCP limite les autorisations pour les entités dans les comptes membres, y compris dans chaque Utilisateur racine d'un compte AWS. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations.

- politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de la séance obtenue sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations, consultez Politiques de séance dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations obtenues sont plus compliquées à comprendre. Pour découvrir la façon dont AWS détermine s'il convient d'autoriser une demande en présence de plusieurs types de politiques, veuillez consulter Logique d'évaluation de politiques dans le Guide de l'utilisateur IAM.

Ressources AWS KMS

Dans AWS KMS, la ressource principale est AWS KMS key. AWS KMS prend également en charge un alias, une ressource indépendante qui fournit un nom convivial pour une clé KMS. Certaines opérations AWS KMS vous permettent d'utiliser un alias pour identifier une clé KMS.

Chaque instance de clé KMS ou d'alias possède un Amazon Resource Name (ARN) unique avec un format standard. Dans les ressources AWS KMS, le nom du service AWS est `kms`.

- AWS KMS key

  Format de nom ARN :

  ```
  arn:AWS partition name:AWS service name:Région AWS:Compte AWS ID:key/key ID
  ```

  Exemple de nom ARN :

```
arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

- Alias

  Format de nom ARN :

  ```
  arn:AWS partition name:AWS service name:Région AWS:Compte AWS
  ID:alias/alias name
  ```

  Exemple de nom ARN :

  ```
  arn:aws:kms:us-west-2:111122223333:alias/example-alias
  ```

AWS KMS fournit un ensemble d'opérations API pour utiliser vos ressources AWS KMS. Pour plus d'informations sur l'identification des clés KMS dans les opérations API AWS Management Console et AWS KMS, veuillez consulter Identifiants clés () KeyId. Pour une liste des opérations AWS KMS, veuillez consulter la AWS Key Management ServiceRéférence API.

Politiques de clé dans AWS KMS

Une politique de clé est une politique de ressources pour une AWS KMS key. Les politiques de clé constituent le principal moyen de contrôler l'accès aux clés KMS. Chaque clé KMS doit avoir exactement une politique de clé. Les instructions dans la politique de clé déterminent qui a l'autorisation d'utiliser la clé KMS et la façon dont ces personnes peuvent l'utiliser. Vous pouvez également utiliser les politiques IAM et les octrois pour contrôler l'accès à la clé KMS, mais chaque clé KMS doit avoir une politique de clé.

Aucun principal AWS, y compris l'utilisateur root ou le créateur de la clé, ne dispose d'autorisations relatives à une clé KMS, sauf s'il est explicitement autorisé, et jamais refusé, dans une politique de clé, une politique IAM ou un octroi.

À moins que la politique de clé ne l'autorise explicitement, vous ne pouvez pas utiliser de politiques IAM pour autoriser l'accès à une clé KMS. Sans autorisation de la politique de clé, les politiques IAM qui octroient des autorisations n'ont aucun effet. (Vous pouvez utiliser une politique IAM pour refuser une autorisation à une clé KMS sans l'autorisation d'une politique de clé.) La politique de clé par défaut active les politiques IAM. Pour activer les politiques IAM dans votre politique de clé, ajoutez l'instruction de politique décrite dans Autorise l'accès au Compte AWS et active les politiques IAM..

Contrairement aux politiques IAM, qui sont mondiales, les politiques de clés sont régionales. Une politique de clé contrôle uniquement l'accès à une clé KMS dans la même région. Elle n'a aucun effet sur les clés KMS des autres régions.

Rubriques

- Création d'une politique de clé
- politique de clé par défaut
- Affichage d'une politique de clé
- Modification d'une politique de clé
- Autorisations pour les services AWS dans les politiques de clé

Création d'une politique de clé

Vous pouvez créer et gérer des politiques clés dans la AWS KMS console, à l'aide d'opérations d'AWS KMSAPI, telles que CreateKeyReplicateKey, et PutKeyPolicy, ou à l'aide d'un AWS CloudFormationmodèle.

Lorsque vous créez une clé KMS dans la console AWS KMS, la console vous guide tout au long des étapes de création d'une politique de clé basée sur la politique de clé par défaut de la console. Lorsque vous utilisez es API `CreateKey` ou `ReplicateKey`, si vous ne spécifiez pas de politique de clé, ces API appliquent la politique de clé par défaut pour les clés créées par programmation. Lorsque vous créez l'API `PutKeyPolicy`, vous devez spécifier une stratégie de clé.

Chaque document de politique peut contenir une ou plusieurs instruction(s) de politique. L'exemple suivant montre un document de politique de clé valide avec une instruction de politique.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Describe the policy statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/Alice"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "kms:KeySpec": "SYMMETRIC_DEFAULT"
        }
      }
    }
  ]
}
```

Rubriques

- Format de politique de clé
- Éléments d'une politique de clé
- Exemple de politique de clé

Format de politique de clé

Un document de politique de clé doit être conforme aux règles suivantes :

- Jusqu'à 32 kilooctets (32 768 octets)
- L'élément `Sid` dans une instruction de politique de clé peut inclure des espaces. (Les espaces sont interdits dans l'élément `Sid` d'un document de politique IAM.)

Un document de politique de clé ne peut contenir que les caractères suivants :

- Caractères ASCII imprimables
- Caractères imprimables du jeu de caractères Basic Latin et du jeu de caractères supplémentaires Latin-1
- Les caractères spéciaux tabulation (`\u0009`), saut de ligne (`\u000A`) et retour chariot (`\u000D`)

Éléments d'une politique de clé

Un document de politique de clé doit disposer des éléments suivants :

Version

Spécifie la version du document de la politique de clé. Définissez la version sur `2012-10-17` (la dernière version).

Instruction

Comprend les instructions de la politique. Un document de politique de clé doit avoir au moins une instruction.

Chaque instruction de politique de clé est composée de six éléments. Les éléments `Effect`, `Principal`, `Action`, et `Resource` sont obligatoires.

Sid

(Facultatif) L'identifiant d'instruction (`Sid`) est une chaîne arbitraire que vous pouvez utiliser pour identifier l'instruction. Le `Sid` dans une politique de clé peut inclure des espaces. (Vous ne pouvez pas inclure d'espaces dans un élément `Sid` de politique IAM.)

Effet

(Requis) Détermine s'il convient d'autoriser ou de rejeter les autorisations figurant dans l'instruction de politique. Les valeurs valides sont `Allow` ou `Deny`. Si vous n'autorisez pas explicitement l'accès à une clé KMS, l'accès est implicitement refusé. Vous pouvez explicitement refuser l'accès à une clé KMS. Vous pouvez le faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une politique différente autorise l'accès.

Principal

(Requis) Le principal est l'identité qui obtient les autorisations figurant dans l'instruction de politique. Vous pouvez spécifier les Comptes AWS, les utilisateurs IAM, les rôles IAM et certains services AWS en tant que principaux dans une politique de clé. Les groupes d'utilisateurs IAM ne constituent un principal valide dans aucun type de politique.

Une valeur astérisque, par exemple `"AWS": "*"`, représente toutes les identités AWS de tous les comptes.

> **Important**
>
> Ne définissez pas le principal sur un astérisque (*) dans une instruction de politique de clé qui autorise des autorisations, sauf si vous utilisez des conditions pour limiter la stratégie de clé. Un astérisque autorise chaque identité dans chaque Compte AWS à utiliser la clé KMS, sauf si une autre instruction de politique la refuse explicitement. Les utilisateurs dans d'autres Comptes AWS peuvent utiliser votre clé KMS dès qu'ils ont les autorisations correspondantes dans leur propre compte.

> **Note**
>
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

Lorsque le principal dans une instruction de politique de clé est un Compte AWS principal exprimé comme `arn:aws:iam::111122223333:root"`, la déclaration de stratégie n'accorde aucune autorisation à un principal IAM. Au lieu de cela, il donne l'autorisation au Compte AWS d'utiliser des politiques IAM pour déléguer les autorisations spécifiées dans la politique de clé. (Un principal au format `arn:aws:iam::111122223333:root"` ne représente pas l'utilisateur racine du compte AWS, malgré l'utilisation de « root » [racine] dans l'identifiant du compte. Cependant, le principal du compte représente le compte et ses administrateurs, y compris l'utilisateur racine du compte.)

Lorsque le principal est un autre Compte AWS ou ses principaux, les autorisations ne sont effectives que lorsque le compte est activé dans la région avec la clé KMS et la politique de clé. Pour plus d'informations sur les régions qui ne sont pas activées par défaut (« Régions d'adhésion »), veuillez consulter Gestion de Régions AWS dans la Références générales AWS.

Pour autoriser un autre Compte AWS ou ses principaux à utiliser une clé KMS, vous devez fournir une autorisation dans une politique de clé et dans une politique IAM de l'autre compte. Pour plus de détails, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

Action

(Requis) Spécifiez les opérations d'API à autoriser ou à rejeter. Par exemple, l'action `kms:Encrypt` correspond à l'opération AWS KMS Encrypt (Chiffrer). Vous pouvez répertorier plusieurs actions dans une instruction de politique. Pour plus d’informations, consultez Référence des autorisations .

Ressource

(Requis) Dans une politique de clé, la valeur de l'élément Ressource est "*", ce qui signifie « cette clé KMS ». L'astérisque ("*") identifie la clé KMS à laquelle la politique de clé est attachée.

> **Note**
>
> Si l'élément `Resource` requis est absent d'une instruction de politique de clé, la déclaration de stratégie n'a aucun effet. Une instruction de politique de clé sans élément `Resource` ne s'applique à aucune clé KMS.
> Lorsqu'il manque un `Resource` élément à une déclaration de politique clé, la AWS KMS console signale correctement une erreur, mais les PutKeyPolicyAPI CreateKeyand réussissent, même si la déclaration de politique est inefficace.

Condition

(Facultatif) Les conditions spécifient les exigences qui doivent être satisfaites pour qu'une politique de clé prenne effet. Les conditions permettent à AWS d'évaluer le contexte d'une demande d'API afin de déterminer si l'instruction de politique s'applique ou non.

Pour spécifier des conditions, vous utilisez des clés de condition prédéfinies. AWS KMS prend en charge les clés de condition globales AWS et les clés de condition AWS KMS. Pour prendre en charge le contrôle d'accès basé sur les attributs (ABAC), AWS KMS fournit des clés de condition qui contrôlent l'accès à une clé KMS en fonction des balises et des alias. Pour plus de détails, consultez ABAC pour AWS KMS.

Le format d'une condition est le suivant :

```
"Condition": {"condition operator": {"condition key": "condition value"}}
```

comme :

```
"Condition": {"StringEquals": {"kms:CallerAccount": "111122223333"}}
```

Pour plus d'informations sur la syntaxe des politiques AWS, veuillez consulter AWS Référence de politique IAM dans le Guide de l'utilisateur IAM.

Exemple de politique de clé

L'exemple suivant illustre une politique de clé complète pour une clé KMS de chiffrement symétrique. Vous pouvez l'utiliser à titre de référence lorsque vous lisez les principaux concepts de stratégie de ce chapitre. Cette politique de clé combine les exemples d'instruction de politique issus de la section précédente politique de clé par défaut en une politique de clé unique qui réalise les opérations suivantes :

- Permet à l'exemple Compte AWS, 111122223333, un accès complet à la clé KMS. Cela permet au compte et à ses administrateurs, y compris l'utilisateur root du compte (pour les urgences), d'utiliser des politiques IAM dans le compte pour autoriser l'accès à la clé KMS.
- Permet au rôle IAM `ExampleAdminRole` d'administrer la clé KMS.
- Permet au rôle IAM `ExampleUserRole` d'utiliser la clé KMS.

```
{
    "Id": "key-consolepolicy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleAdminRole"
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
```

```
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
```

```
    ]
}
```

politique de clé par défaut

Lorsque vous créez une clé KMS, vous pouvez spécifier la politique de clé pour la nouvelle clé KMS. Si vous n'en fournissez pas, AWS KMS en crée une pour vous. La politique de clé par défaut que AWS KMS utilise diffère selon que vous créez la clé dans la console AWS KMS ou que vous utilisez l'API AWS KMS.

politique de clé par défaut lorsque vous créez une clé KMS par programmation

Lorsque vous créez une clé KMS par programmation avec l'API AWS KMS (y compris à l'aide des kits SDK AWS, l'AWS Command Line Interface ou AWS Tools for PowerShell) et que vous ne spécifiez pas de politique de clé, AWS KMS applique une politique de clé par défaut très simple. Cette politique de clé par défaut comporte une instruction de politique qui accorde au Compte AWS propriétaire de la clé KMS l'autorisation d'utiliser les politiques IAM pour autoriser l'accès à toutes les opérations AWS KMS sur la clé KMS. Pour plus d'informations sur cette instruction de politique, consultez la rubrique Autorise l'accès au Compte AWS et active les politiques IAM..

politique de clé par défaut lorsque vous créez une clé KMS à l'aide de laAWS Management Console

Lorsque vous créez une clé KMS avec la AWS Management Console, la politique de clé commence par l'instruction de politique qui autorise l'accès au Compte AWS et active les politiques IAM. La console ajoute ensuite une instruction d'administrateur de clé, une instruction d'utilisateur de clé et (pour la plupart des types de clé) une instruction qui permet aux principals d'utiliser la clé KMS avec d'autres services AWS. Vous pouvez utiliser les fonctions de la console AWS KMS pour spécifier les utilisateurs et rôles IAM, et pour désigner les Comptes AWS qui sont des administrateurs clés et ceux qui sont des utilisateurs clés (ou les deux).

Autorisations

- Autorise l'accès au Compte AWS et active les politiques IAM.
- Autorise les administrateurs de clés à administrer la clé KMS
- Autorise les utilisateurs de clés à utiliser la clé KMS.
  - Permet aux utilisateurs de clés d'utiliser une clé KMS pour les opérations de chiffrement
  - Permet aux utilisateurs de clé d'utiliser la clé KMS avec les services AWS.

Autorise l'accès au Compte AWS et active les politiques IAM.

L'instruction de politique de clé par défaut suivante est extrêmement importante.

- Elle offre au Compte AWS propriétaire de la clé KMS l'accès complet à celle-ci.

  Contrairement aux autres politiques de ressources AWS, une politique de clé AWS KMS n'accorde pas automatiquement l'autorisation au compte ou à l'une de ses identités. Pour accorder l'autorisation aux administrateurs de compte, la politique de clé doit inclure une instruction explicite qui fournit cette autorisation, comme celle-ci.
- Celle permet au compte d'utiliser des politiques IAM pour autoriser l'accès à la clé KMS, en plus de la politique de clé.

  Sans cette autorisation, les politiques IAM qui autorisent l'accès à la clé sont inefficaces, bien que celles qui refusent l'accès à la clé soient toujours efficaces.
- Cela réduit le risque que la clé devienne ingérable en accordant une autorisation de contrôle d'accès aux administrateurs du compte, y compris l'utilisateur racine du compte, qui ne peut pas être supprimé.

L'instruction de politique de clé suivante est la politique de clé par défaut complète pour les clés KMS créées par programmation. Il s'agit de la première instruction de politique dans la politique de clé par défaut pour les clés KMS créées dans la console AWS KMS.

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
   },
  "Action": "kms:*",
  "Resource": "*"
}
```

Autorise les politiques IAM à autoriser l'accès à la clé KMS.

> L'instruction de politique de clé présentée ci-dessus accorde au Compte AWS propriétaire de la clé l'autorisation d'utiliser les politiques IAM, ainsi que les politiques de clé, pour autoriser toutes les actions (`kms:*`) sur la clé KMS.

Le principal dans cette instruction de politique de clé est le principal du compte, qui est représenté par un ARN au format suivant : `arn:aws:iam::account-id:root`. Le principal du compte représente le compte AWS et ses administrateurs.

Lorsque le principal d'une instruction de politique de clé correspond au principal du compte, cette instruction n'autorise aucun principal IAM à utiliser la clé KMS. Au lieu de cela, elle permet au compte d'utiliser des politiques IAM pour déléguer les autorisations spécifiées dans l'instruction de politique. Cette instruction de politique de clé par défaut permet au compte d'utiliser des politiques IAM pour déléguer l'autorisation pour toutes les actions (`kms:*`) sur la clé KMS.

réduit le risque que la clé KMS devienne ingérable.

Contrairement aux autres politiques de ressources AWS, une politique de clé AWS KMS n'accorde pas automatiquement l'autorisation au compte ou à l'un de ses principaux. Pour accorder l'autorisation à un principal, y compris le principal du compte, vous devez utiliser une instruction de politique de clé qui fournit explicitement l'autorisation. Vous n'êtes pas l'obligation de donner accès à la clé KMS au principal du compte ou à tout autre principal. Cependant, donner accès au principal du compte vous permet d'éviter que la clé ne devienne ingérable.

Par exemple, supposons que vous créez une politique de clé qui donne à un seul utilisateur l'accès à la clé KMS. Si vous supprimez ensuite cet utilisateur, la clé devient ingérable et vous devez contacter le support AWS pour retrouver l'accès à la clé KMS.

L'instruction de politique de clé présentée ci-dessus donne l'autorisation de contrôler la clé au principal du compte, qui représente le Compte AWS et ses administrateurs, y compris l'utilisateur racine du compte. L'utilisateur racine du compte est le seul principal qui ne peut pas être supprimé, sauf si vous supprimez le Compte AWS. Conformément aux bonnes pratiques IAM, il est déconseillé d'agir au nom de l'utilisateur racine du compte, sauf en cas d'urgence. Toutefois, vous devrez peut-être agir en tant qu'utilisateur racine du compte si vous supprimez tous les autres utilisateurs et rôles ayant accès à la clé KMS.

Autorise les administrateurs de clés à administrer la clé KMS

La politique de clé par défaut créée par la console vous permet de choisir des utilisateurs et des rôles IAM dans le compte et d'en faire des administrateurs de clé. Cette instruction est appelée l'instruction des administrateurs de clé. Les administrateurs de clés ont les autorisations nécessaires pour gérer la clé KMS, mais n'ont pas d'autorisations pour utiliser la clé KMS dans des opérations de chiffrement. Vous pouvez ajouter des utilisateurs et des rôles IAM à la liste des administrateurs de clés lorsque vous créez la clé KMS dans la vue par défaut ou la vue de la politique.

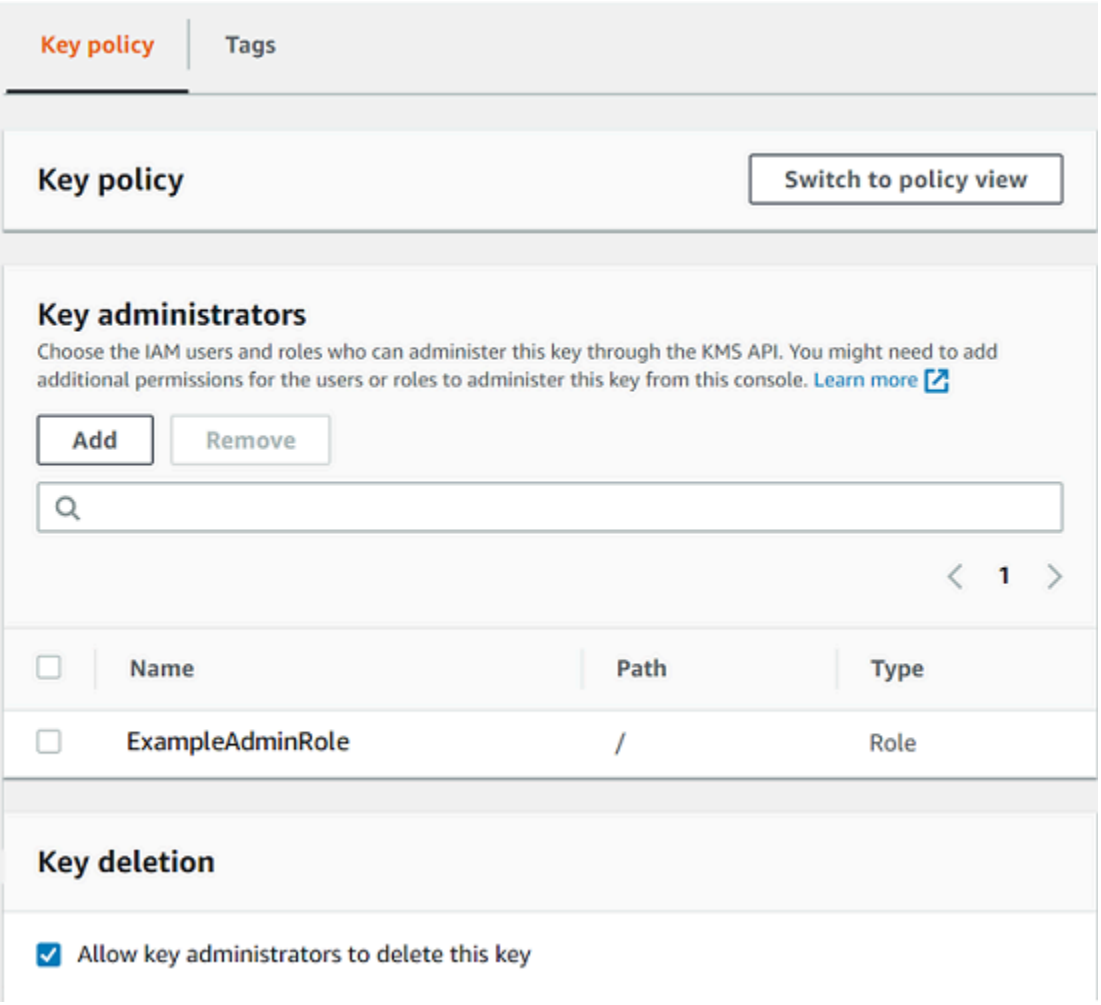
Warning

Étant donné que les administrateurs de clés ont l'autorisation de modifier la politique de clé et de créer des octrois, ils peuvent se donner à eux-mêmes et à d'autres des autorisations AWS KMS non spécifiées dans cette politique.
Les principaux qui ont l'autorisation de gérer les balises et les alias peuvent également contrôler l'accès à une clé KMS. Pour plus de détails, consultez ABAC pour AWS KMS.

Note

Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

L'exemple suivant montre l'instruction des administrateurs de clé dans la vue par défaut de la console AWS KMS.



Voici un exemple d'instruction des administrateurs de clés dans la vue de politique de la console AWS KMS. Cette instruction des administrateurs de clés concerne une clé KMS de chiffrement symétrique à région unique.

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS":"arn:aws:iam::111122223333:role/ExampleAdminRole"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
```

```
    "kms:Get*",
    "kms:Delete*",
    "kms:TagResource",
    "kms:UntagResource",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

L'instruction par défaut des administrateurs de clés pour la clé KMS la plus courante, une clé KMS de chiffrement symétrique à région unique, octroie les autorisations suivantes. Pour plus d'informations sur ces autorisations, veuillez consulter la AWS KMS autorisations.

Lorsque vous utilisez la console AWS KMS pour créer une clé KMS, la console ajoute les utilisateurs et les rôles que vous spécifiez à l'élément `Principal` dans l'instruction des administrateurs de clé.

Un grand nombre de ces autorisations contiennent le caractère générique (*), qui octroie toutes les autorisations commençant par le verbe spécifié. Par conséquent, lorsque AWS KMS ajoute de nouvelles opérations d'API, les administrateurs de clés sont automatiquement autorisés à les utiliser. Il n'est pas nécessaire de mettre à jour vos politiques de clé pour inclure les nouvelles opérations. Si vous préférez limiter vos administrateurs de clés à un ensemble fixe d'opérations d'API, vous pouvez modifier votre politique de clé.

`kms:Create*`

Autorise `kms:CreateAlias` et `kms:CreateGrant`. (L'autorisation `kms:CreateKey` n'est valide que dans une politique IAM.)

`kms:Describe*`

Autorise `kms:DescribeKey` L'autorisation `kms:DescribeKey` est nécessaire pour afficher la page des détails d'une clé KMS dans la AWS Management Console.

`kms:Enable*`

Autorise `kms:EnableKey` Pour les clés KMS de chiffrement symétriques, elle autorise également `kms:EnableKeyRotation`.

`kms:List*`

Autorise `kms:ListGrants`, `kms:ListKeyPolicies` et `kms:ListResourceTags`. (Les autorisations `kms:ListAliases` et `kms:ListKeys`, requises pour afficher les clés KMS dans la AWS Management Console, sont valides uniquement dans les politiques IAM.)

`kms:Put*`

Autorise `kms:PutKeyPolicy` Cette autorisation autorise les administrateurs de clés à modifier la politique de cette clé KMS.

`kms:Update*`

Autorise `kms:UpdateAlias` et `kms:UpdateKeyDescription`. Pour les clés multi-région, elle autorise `kms:UpdatePrimaryRegion` sur cette clé KMS.

`kms:Revoke*`

Autorise `kms:RevokeGrant` qui permet aux administrateurs de clés de supprimer un octroi, même s'ils ne sont pas un principal de retrait dans l'octroi.

`kms:Disable*`

Autorise `kms:DisableKey` Pour les clés KMS de chiffrement symétriques, elle autorise également `kms:DisableKeyRotation`.

`kms:Get*`

Autorise `kms:GetKeyPolicy` et `kms:GetKeyRotationStatus`. Pour les clés KMS avec des éléments de clé importés, elle autorise `kms:GetParametersForImport`. Pour les clés KMS asymétriques, elle autorise `kms:GetPublicKey`. L'autorisation `kms:GetKeyPolicy` est nécessaire pour afficher la politique de clé KMS dans la AWS Management Console.

`kms:Delete*`

Autorise `kms:DeleteAlias` Pour les clés avec des éléments de clé importés, elle autorise `kms:DeleteImportedKeyMaterial`. Notez que l'autorisation `kms:Delete*` ne permet pas aux administrateurs de clés de supprimer la clé KMS (`ScheduleKeyDeletion`).

`kms:TagResource`

Autorise `kms:TagResource`, qui permet aux administrateurs de clés d'ajouter des identifications à la clé KMS. Étant donné que les balises peuvent être également utilisées pour contrôler l'accès à la clé KMS, cette autorisation peut permettre aux administrateurs d'autoriser ou de refuser l'accès à la clé KMS. Pour plus de détails, consultez ABAC pour AWS KMS.

`kms:UntagResource`

Autorise `kms:UntagResource`, qui permet aux administrateurs de clés de supprimer les balises de la clé KMS. Étant donné que les balises peuvent être utilisées pour contrôler l'accès à la clé, cette autorisation peut permettre aux administrateurs d'autoriser ou de refuser l'accès à la clé KMS. Pour plus de détails, consultez ABAC pour AWS KMS.

`kms:ScheduleKeyDeletion`

Autorise `kms:ScheduleKeyDeletion`, qui permet aux administrateurs de clés de supprimer cette clé KMS. Pour supprimer cette autorisation, désélectionnez l'option Autoriser les administrateurs de clé à supprimer cette clé.

`kms:CancelKeyDeletion`

Autorise `kms:CancelKeyDeletion`, qui permet aux administrateurs de clés d'annuler la suppression de cette clé KMS. Pour supprimer cette autorisation, désélectionnez l'option Autoriser les administrateurs de clé à supprimer cette clé.

AWS KMS ajoute les autorisations suivantes à l'instruction par défaut des administrateurs de clés lorsque vous créez clés spéciales.

`kms:ImportKeyMaterial`

L'autorisation `kms:ImportKeyMaterial` permet aux administrateurs de clés d'importer des éléments de clé dans la clé KMS. Cette autorisation est incluse dans la politique de clé uniquement lorsque vous créez une clé KMS sans élément de clé.

`kms:ReplicateKey`

L'autorisation `kms:ReplicateKey` permet aux administrateurs de clés de créer un réplica d'une clé primaire multi-région dans une autre région AWS. Cette autorisation est incluse dans la politique de clé uniquement lorsque vous créez une clé primaire ou un réplica multi-région.

`kms:UpdatePrimaryRegion`

L'autorisation `kms:UpdatePrimaryRegion` permet aux administrateurs de clés de remplacer une clé de réplica multi-région par une clé primaire multi-région. Cette autorisation est incluse dans la politique de clé uniquement lorsque vous créez une clé primaire ou un réplica multi-région.

Autorise les utilisateurs de clés à utiliser la clé KMS.

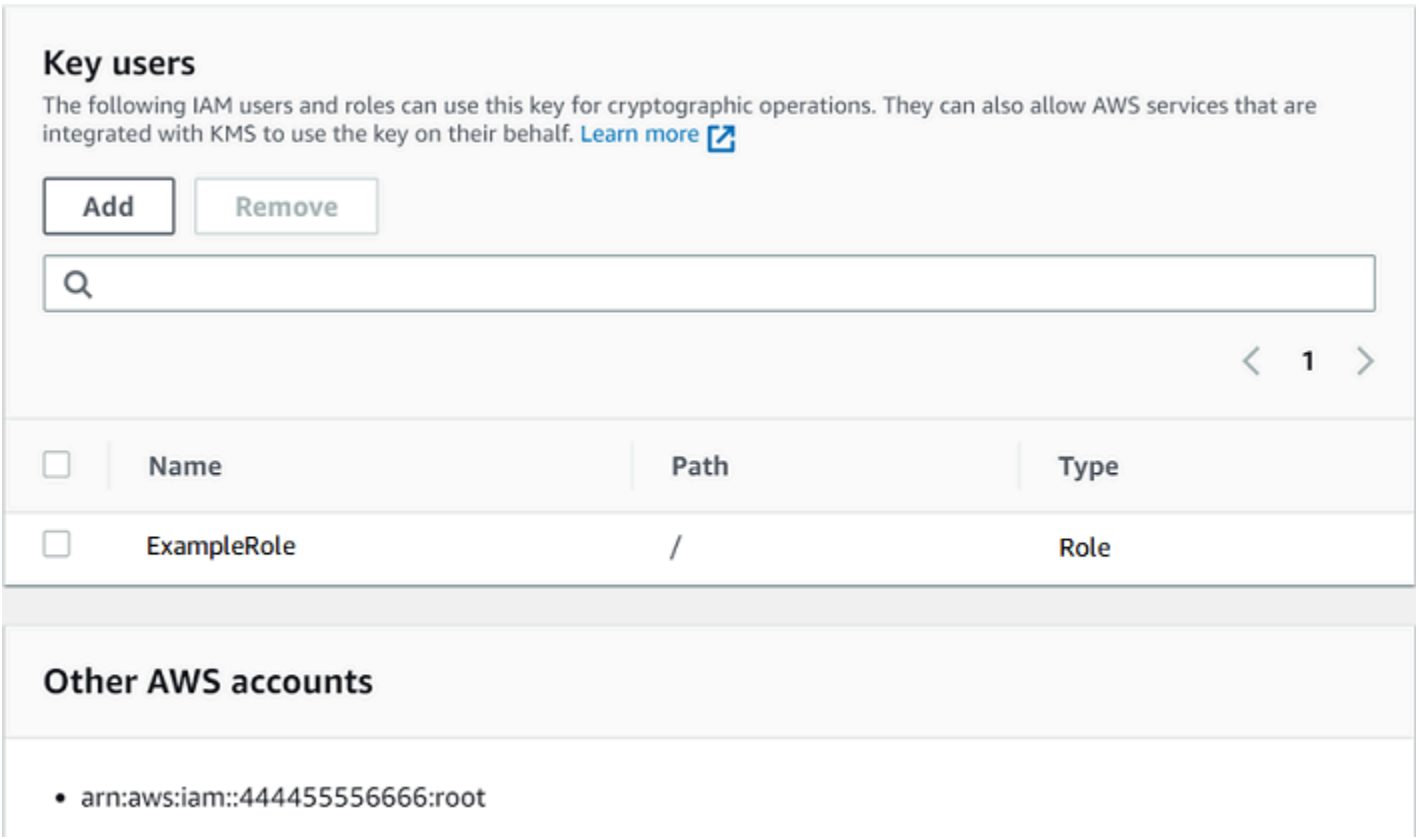
La politique de clé par défaut créée par la console pour les clés KMS vous permet de choisir des utilisateurs et rôles IAM du compte ainsi que des Comptes AWS externes, et d'en faire des utilisateurs de clé.

La console ajoute deux instructions de politique à la politique de clé pour les utilisateurs de clés.

- Utilisez la clé KMS directement – La première instruction de politique de clé donne aux utilisateurs des clés l'autorisation d'utiliser la clé KMS directement pour toutes les opérations de chiffrement prises en charge pour ce type de clé KMS.
- Utilisez la clé KMS avec les services AWS – La deuxième instruction de politique permet aux utilisateurs de clés d'autoriser les services AWS intégrés à AWS KMS à utiliser la clé KMS en leur nom pour protéger les ressources, telles que les compartiments Amazon S3 et les tables Amazon DynamoDB.

Vous pouvez ajouter des utilisateurs IAM, des rôles IAM et d'autres Comptes AWS à la liste des utilisateurs de clé lorsque vous créez la clé KMS. Vous pouvez aussi modifier la liste avec la vue par défaut de la console pour les politiques de clé, comme illustré dans l'image suivante. La vue par défaut pour les politiques de clé est sur la page des détails de la clé. Pour plus d'informations sur l'autorisation accordée à des utilisateurs d'autres Comptes AWS d'utiliser la clé KMS, veuillez consulter Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

> **Note**
>
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.



Les instructions par défaut des utilisateurs de clé d'une clé symétrique à région unique octroient les autorisations suivantes. Pour plus d'informations sur ces autorisations, veuillez consulter la AWS KMS autorisations.

Lorsque vous utilisez la console AWS KMS pour créer une clé KMS, la console ajoute les utilisateurs et les rôles que vous spécifiez à l'élément `Principal` dans chaque instruction des utilisateurs de clé.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
```

```
  ],
  "Resource": "*"
},
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

Permet aux utilisateurs de clés d'utiliser une clé KMS pour les opérations de chiffrement

Les utilisateurs de clés ont l'autorisation d'utiliser la clé KMS directement dans toutes les opérations de chiffrement prises en charge par la clé KMS. Ils peuvent également utiliser l'DescribeKeyopération pour obtenir des informations détaillées sur la clé KMS dans la AWS KMS console ou en utilisant les opérations AWS KMS d'API.

Par défaut, la console AWS KMS ajoute des instructions des utilisateur de clés comme celles des exemples suivants à la politique de clé par défaut. Étant donné qu'ils prennent en charge différentes opérations d'API, les actions des instructions de politique pour les clés KMS de chiffrement symétriques, les clés KMS HMAC, les clés KMS asymétriques pour le chiffrement de clé publique et les clés KMS asymétriques pour la signature et la vérification sont légèrement différentes.

Clés KMS de chiffrement symétriques

La console ajoute l'instruction suivante à la politique de clé pour les clés KMS de chiffrement symétriques.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
```

```
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey",
    "kms:Encrypt",
    "kms:GenerateDataKey*",
    "kms:ReEncrypt*"
  ],
  "Resource": "*"
}
```

Clés KMS HMAC

La console ajoute l'instruction suivante à la politique de clé pour les clés KMS HMAC.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateMac",
    "kms:VerifyMac"
  ],
  "Resource": "*"
}
```

Clés CMK asymétriques pour le chiffrement de clé publique

La console ajoute l'instruction suivante à la politique de clé pour les clés KMS asymétriques avec Encrypt and decrypt (Chiffrer et déchiffrer) comme utilisation des clés.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey",
```

```
    "kms:GetPublicKey"
  ],
  "Resource": "*"
}
```

Clés CMK asymétriques pour la signature et la vérification

La console ajoute l'instruction suivante à la politique de clé pour les clés KMS asymétriques avec Sign and verify (Signer et vérifier) comme utilisation des clés.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:Sign",
    "kms:Verify"
  ],
  "Resource": "*"
}
```

Les actions de ces instructions donnent aux utilisateurs de clé les autorisations suivantes.

`kms:Encrypt`

Permet aux utilisateurs de clé de chiffrer des données avec cette clé KMS.

`kms:Decrypt`

Permet aux utilisateurs de clé de déchiffrer des données avec cette clé KMS.

`kms:DescribeKey`

Permet aux utilisateurs de clé d'obtenir des informations sur cette clé KMS, y compris ses identificateurs, sa date de création, son état, etc. Les utilisateurs de clé peuvent également afficher des détails sur la clé KMS dans la console AWS KMS.

`kms:GenerateDataKey*`

Permet aux utilisateurs de clé de demander une paires de clés de données symétriques ou asymétriques pour les opérations de chiffrement côté client. La console utilise

le caractère générique * pour représenter l'autorisation pour les opérations d'API suivantes : GenerateDataKey, GenerateDataKeyWithoutPlaintextGenerateDataKeyPair, et GenerateDataKeyPairWithoutPlaintext. Ces autorisations sont valides uniquement sur les clés KMS symétriques qui chiffrent les clés de données.

km : GenerateMac

Permet aux utilisateurs de clés d'utiliser une clé KMS HMAC pour générer une balise HMAC.

km : GetPublicKey

Permet aux utilisateurs de clé de télécharger la clé publique de la clé KMS asymétrique. Les parties avec lesquelles vous partagez cette clé publique peuvent chiffrer les données en dehors de AWS KMS. Cependant, ces textes chiffrés ne peuvent être déchiffrés qu'en appelant l'opération Decrypt dans AWS KMS.

km : ReEncrypt *

Permet aux utilisateurs de clé de rechiffrer les données initialement chiffrées avec cette clé KMS, ou d'utiliser cette clé KMS pour rechiffrer des données précédemment chiffrées. L'ReEncryptopération nécessite l'accès aux clés KMS source et de destination. Pour ce faire, vous pouvez accorder l'autorisation `kms:ReEncryptFrom` sur la clé KMS source et l'autorisation `kms:ReEncryptTo` sur la clé KMS de destination. Cependant, par souci de simplicité, la console autorise `kms:ReEncrypt*` (avec le caractère * générique) sur les deux clés KMS.

kms:Sign

Permet aux utilisateurs de clé de signer des messages avec cette clé KMS.

kms:Verify

Permet aux utilisateurs de clé de vérifier les signatures avec cette clé KMS.

km : VerifyMac

Permet aux utilisateurs de clés d'utiliser une clé KMS HMAC pour vérifier une balise HMAC.

Permet aux utilisateurs de clé d'utiliser la clé KMS avec les services AWS.

La politique de clé par défaut dans la console donne également aux utilisateurs clés les autorisations d'octroi dont ils ont besoin pour protéger leurs données dans les services AWS qui utilisent des octrois. Les services AWS utilisent souvent des octrois pour obtenir une autorisation spécifique et limitée d'utiliser une clé KMS.

Cette instruction de politique de clé autorise l'utilisateur de la clé à créer, afficher et révoquer des octrois sur la clé KMS, mais uniquement lorsque la demande provient d'un service AWS intégré à AWS KMS. La condition de GrantIsFor AWSResource politique kms : ne permet pas à l'utilisateur d'appeler directement ces opérations de subvention. Lorsque l'utilisateur de la clé l'autorise, un service AWS peut créer une autorisation au nom de l'utilisateur qui permet au service d'utiliser la clé KMS pour protéger les données de l'utilisateur.

Les utilisateurs de clé ont besoin de ces autorisations pour utiliser leur clé KMS avec des services intégrés, mais ces autorisations ne sont pas suffisantes. Les utilisateurs de clés ont également besoin d'une autorisation pour utiliser les services intégrés. Pour plus d'informations sur l'attribution aux utilisateurs de l'accès à un service AWS qui s'intègre à AWS KMS, consultez la documentation relative au service intégré.

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

Par exemple, les utilisateurs de clés peuvent utiliser ces autorisations sur la clé KMS de la manière suivante.

- Utilisez cette clé KMS avec Amazon Elastic Block Store (Amazon EBS) et Amazon Elastic Compute Cloud (Amazon EC2) pour attacher un volume EBS chiffré à une instance EC2. L'utilisateur de clé accorde implicitement à Amazon EC2 l'autorisation d'utiliser la clé KMS pour attacher le volume chiffré à l'instance. Pour plus d’informations, consultez Comment Amazon Elastic Block Store (Amazon EBS) utilise AWS KMS.
- Utilisez cette clé KMS avec Amazon Redshift pour lancer un cluster chiffré. L'utilisateur de clé accorde implicitement à Amazon Redshift l'autorisation d'utiliser la clé KMS pour lancer le cluster chiffré et créer des instantanés chiffrés. Pour plus d’informations, consultez Comment d'Amazon Redshift utilise AWS KMS.

- Utiliser cette clé KMS avec d'autres services AWS intégrés à AWS KMS qui utilisent des octrois, pour créer, gérer ou utiliser des ressources chiffrées avec ces services.

La politique de clé par défaut permet aux utilisateurs clés de déléguer leur autorisation d'octroi à tous les services intégrés qui utilisent des octrois. Cependant, vous pouvez créer une politique de clé personnalisée qui limite l'autorisation aux services AWS spécifiés. Pour plus d'informations, reportez-vous à la clé de condition km : ViaService.

Affichage d'une politique de clé

Vous pouvez consulter la politique clé d'une clé gérée par le AWS KMS client ou d'une clé intégrée Clé gérée par AWSà votre compte en utilisant l'opération AWS Management Console ou l'GetKeyPolicyopération dans l'AWS KMSAPI. Vous ne pouvez pas utiliser ces techniques pour afficher la politique d'une clé KMS dans un autre Compte AWS.

Pour en savoir plus sur les politiques de clé AWS KMS, consultez Politiques de clé dans AWS KMS. Pour savoir comment déterminer quels sont les utilisateurs et rôles qui ont accès à une clé KMS, veuillez consulter the section called “Détermination de l'accès”.

Rubriques

- Affichage d'une politique de clé (console)
- Affichage d'une politique de clé (API AWS KMS)

Affichage d'une politique de clé (console)

Les utilisateurs autorisés peuvent afficher la politique de clé d'une Clé gérée par AWS ou d'une clé gérée par le client dans l'onglet Politique de clé de la AWS Management Console.

Pour afficher la politique clé d'une clé KMS dans leAWS Management Console, vous devez disposer des GetKeyPolicy autorisations kms : DescribeKey, kms : et kms :. ListAliases

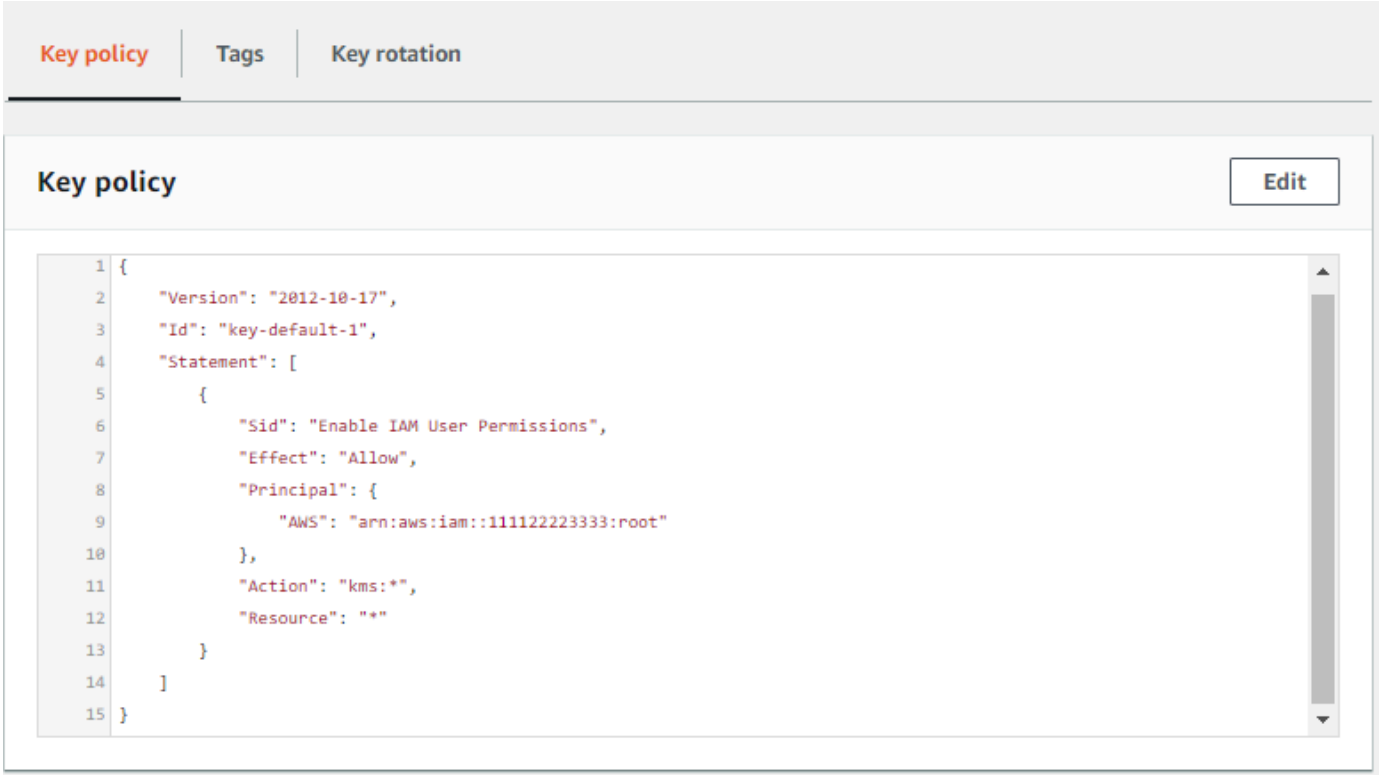
1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3.

Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client.

4. Dans la liste des clés KMS, choisissez l'alias ou l'ID de clé de la clé KMS que vous souhaitez examiner.
5. Choisissez l'onglet Politique de clé.

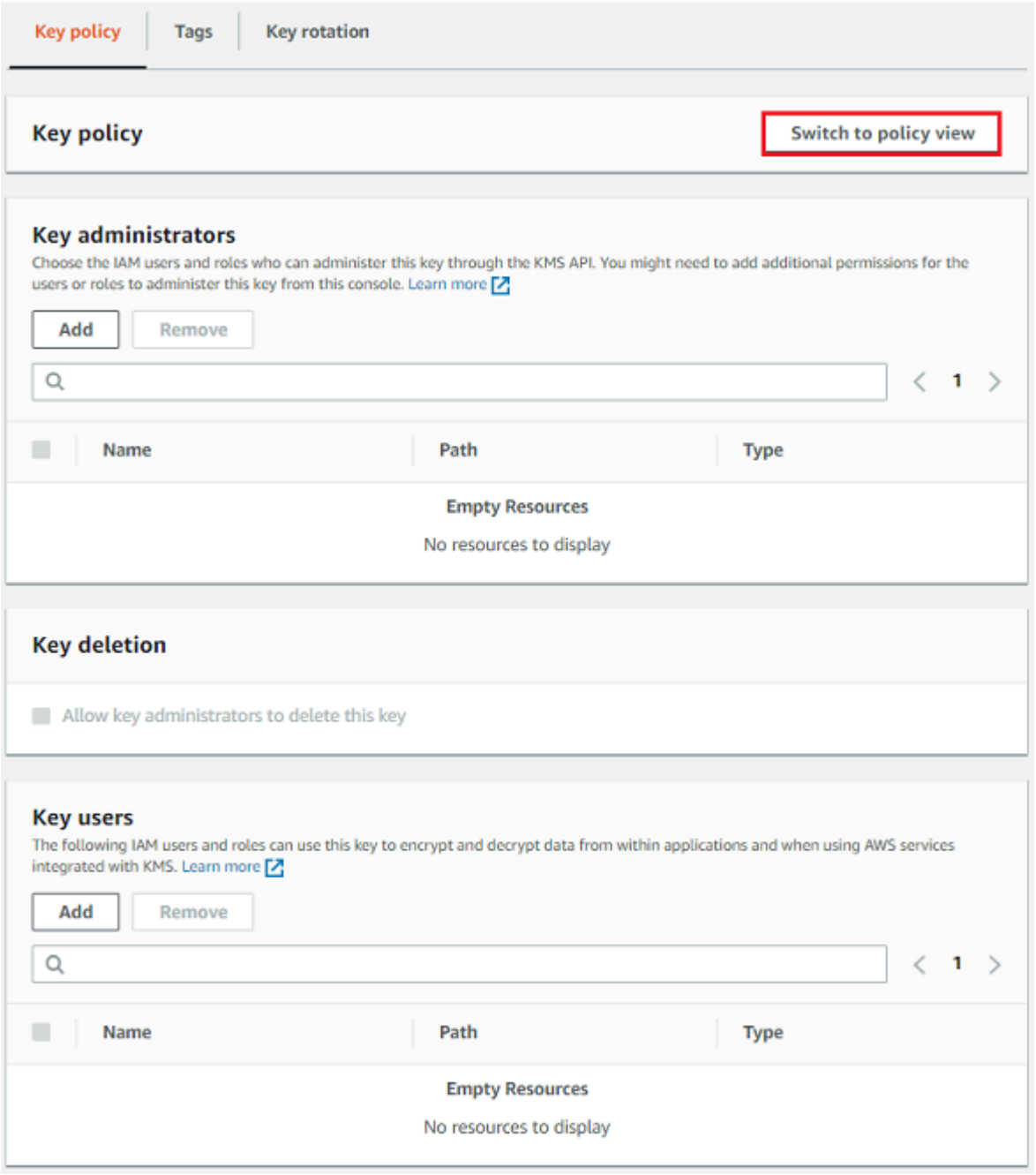
   Dans l'onglet Politique de clé, vous pouvez voir le document de politique de clé. Il s'agit d'une vue de politique. Dans les instructions de politique de clé, vous pouvez voir les principaux qui sont autorisés à accéder à la clé KMS par la politique de clé, ainsi que les actions qu'ils peuvent effectuer.

   L'exemple suivant montre la vue de la politique de clé par défaut.



   Ou, si vous avez créé la clé KMS dans la AWS Management Console, vous verrez la vue par défaut avec des sections pour les Administrateurs de clé, la Suppression de clé et les Utilisateurs de clé. Pour consulter le document de politique de clé, choisissez Passer à la vue de politique.

   L'exemple suivant montre la vue par défaut de la politique de clé par défaut.



Affichage d'une politique de clé (API AWS KMS)

Pour obtenir la politique de clé pour une clé KMS dans votreCompte AWS, utilisez l'GetKeyPolicyopération dans l'AWS KMSAPI. Vous ne pouvez pas utiliser cette opération pour afficher une politique de clé d'un autre compte.

L'exemple suivant utilise la get-key-policycommande contenue dans le AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel AWS SDK pour effectuer cette demande.

Notez que le paramètre `PolicyName` est obligatoire, même si `default` est sa seule valeur valide. En outre, cette commande demande une sortie en texte, plutôt qu'en JSON, pour le rendre plus facile à afficher.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de clé par un identifiant valide provenant de votre compte.

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default --output text
```

La réponse doit être similaire à la suivante, qui renvoie la politique de clé par défaut.

```
{
  "Version" : "2012-10-17",
  "Id" : "key-consolepolicy-3",
  "Statement" : [ {
    "Sid" : "Enable IAM User Permissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

Modification d'une politique de clé

Vous pouvez modifier la politique de clé d'une clé KMS dans votre ordinateur en Compte AWS utilisant l'PutKeyPolicyopération AWS Management Console ou. Vous ne pouvez pas utiliser ces techniques pour modifier la politique clé d'une clé KMS d'un autre Compte AWS.

Lors de la modification d'une politique de clé, gardez à l'esprit les règles suivantes :

- Vous pouvez afficher la politique de clé d'une Clé gérée par AWS ou d'une clé KMS par le client, mais vous ne pouvez modifier que la politique de clé d'une clé KMS géré par le client. Les politiques des Clés gérées par AWS sont créées et gérées par le service AWS qui a créé la clé KMS dans votre compte. Vous ne pouvez pas afficher ou modifier la politique de clé pour une Clé détenue par AWS.

- Vous pouvez ajouter ou supprimer des utilisateurs IAM, des rôles IAM et des Comptes AWS dans la politique de clé, ainsi que modifier les actions autorisées ou refusées pour ces principaux. Pour plus d'informations sur les moyens de spécifier des principaux et des autorisations dans une politique de clé, consultez la page Politiques de clé.
- Vous ne pouvez pas ajouter de groupes IAM à une politique de clé, mais vous pouvez ajouter plusieurs utilisateurs IAM et rôles IAM. Pour plus d'informations, consultez Attribution à plusieurs principaux IAM de l'autorisation d'accès à une clé KMS.
- Lorsque vous ajoutez des Comptes AWS externes à une politique de clé, vous devez également utiliser des politiques IAM dans les comptes externes pour accorder des autorisations aux utilisateurs, aux groupes et aux rôles IAM de ces comptes. Pour plus d'informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.
- Le document de politique de clé obtenu ne doit pas comporter plus de 32 Ko (32 768 octets).

Rubriques

- Comment modifier une politique de clé
- Attribution à plusieurs principaux IAM de l'autorisation d'accès à une clé KMS

Comment modifier une politique de clé

Vous pouvez modifier une politique de clé de trois façons différentes, chacune d'elles étant expliquée dans les sections suivantes.

Rubriques

- Utilisation de la vue par défaut d'AWS Management Console
- Utilisation de la vue de politique d'AWS Management Console
- Utilisation de l'API AWS KMS

Utilisation de la vue par défaut d'AWS Management Console

Vous pouvez utiliser la console pour modifier une politique de clé à l'aide d'une interface graphique appelée vue par défaut.

Si les étapes suivantes ne correspondent pas à ce que vous voyez dans la console, cela peut signifier que cette politique n'a pas été créée par la console. Ou cela peut signifier que la politique de clé a été modifiée d'une façon que la vue par défaut de la console ne prend pas en charge. Dans ce

cas, suivez les étapes de la section Utilisation de la vue de politique d'AWS Management Console ou Utilisation de l'API AWS KMS.

1. Affichez la politique de clé d'une clé gérée par le client comme indiqué dans Affichage d'une politique de clé (console). (Vous ne pouvez pas modifier les politiques de clé des Clés gérées par AWS.)
2. Décidez ce qu'il convient de modifier.

   - Pour ajouter ou supprimer des administrateurs de clé, et pour autoriser ou non ces administrateurs de clé à supprimer la clé KMS, utilisez les contrôles de la section Administrateurs de clé de la page. Les administrateurs de clé gèrent la clé KMS, y compris son activation et sa désactivation, la définition de la politique de clé, et l'activation de la rotation des clés.
   - Pour ajouter ou supprimer des utilisateurs de clé, et pour autoriser ou non les comptes Comptes AWS externes à utiliser la clé KMS, utilisez les contrôles de la section Key users (Utilisateurs de clé) de la page. Les utilisateurs de clé peuvent utiliser la clé KMS dans les opérations de chiffrement, telles que le chiffrement, le déchiffrement, le rechiffrement et la génération de clés de données.

Utilisation de la vue de politique d'AWS Management Console

Vous pouvez utiliser la console pour modifier un document de politique de clé à l'aide de la vue de politique de la console.

1. Affichez la politique de clé d'une clé gérée par le client comme indiqué dans Affichage d'une politique de clé (console). (Vous ne pouvez pas modifier les politiques de clé des Clés gérées par AWS.)
2. Dans la section Politique de clé, choisissez Passer à la vue de politique.
3. Modifiez le document de politique de clé, puis choisissez Enregistrer les modifications.

Utilisation de l'API AWS KMS

Vous pouvez utiliser cette PutKeyPolicyopération pour modifier la politique de clé d'une clé KMS dans votreCompte AWS. Vous ne pouvez pas utiliser cette API sur une clé KMS d'un autre Compte AWS.

1. Utilisez cette GetKeyPolicyopération pour obtenir le document de stratégie clé existant, puis enregistrez le document de stratégie clé dans un fichier. Pour obtenir un exemple de code dans plusieurs langages de programmation, veuillez consulter Obtention d'une politique de clé.
2. Ouvrez le document de politique de clé dans votre éditeur de texte préféré, modifiez le document de politique de clé, puis enregistrez le fichier.
3. Utilisez cette PutKeyPolicyopération pour appliquer le document de politique clé mis à jour à la clé KMS. Pour obtenir un exemple de code dans plusieurs langages de programmation, veuillez consulter Définition d'une politique de clé.

Pour un exemple de copie d'une politique clé d'une clé KMS à une autre, consultez l'GetKeyPolicy exemple dans le manuel de référence des AWS CLI commandes.

Attribution à plusieurs principaux IAM de l'autorisation d'accès à une clé KMS

Les groupes IAM ne sont pas des principaux valides dans une politique de clé. Pour autoriser plusieurs utilisateurs et rôles à accéder à une clé KMS, effectuez l'une des opérations suivantes :

- Utilisez un rôle IAM comme principal dans la politique de clé. Plusieurs utilisateurs autorisés peuvent assumer ce rôle selon les besoins. Pour plus d'informations, consultez Rôles IAM dans le Guide de l'utilisateur IAM.

  Rien ne vous empêche d'associer plusieurs utilisateurs IAM à une politique de clé, mais cette pratique est déconseillée, car elle vous oblige à mettre à jour la politique de clé chaque fois que la liste des utilisateurs autorisés change. En outre, les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.
- Utilisez une politique IAM pour accorder une autorisation à un groupe IAM. Pour ce faire, assurez-vous que la politique de clé contient la déclaration qui permet aux politiques IAM d'autoriser l'accès à la clé KMS, créez une politique IAM qui autorise l'accès à la clé KMS, puis attribuez cette politique à un groupe IAM dans lequel figurent les utilisateurs IAM autorisés. Grâce à cette approche, vous n'avez pas besoin de modifier de politiques lorsque la liste des utilisateurs autorisés change. Au lieu de cela, il vous suffit d'ajouter ou de supprimer ces utilisateurs à partir du groupe IAM approprié. Pour plus d'informations, consultez Groupes d'utilisateurs IAM dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur la manière dont les politiques de clé AWS KMS et les politiques IAM fonctionnent ensemble, veuillez consulter Résolution des problèmes de clé d'accès.

Autorisations pour les services AWS dans les politiques de clé

De nombreux services AWS utilisent AWS KMS keys pour protéger les ressources qu'ils gèrent. Lorsqu'un service utilise Clés détenues par AWS ou Clés gérées par AWS, le service établit et gère les politiques de clé de ces clés KMS.

Toutefois, lorsque vous utilisez une clé gérée par le client avec un service AWS vous définissez et gérez la politique de clé. Cette politique de clé doit accorder au service les autorisations minimales dont il a besoin pour protéger la ressource en votre nom. Nous vous recommandons de respecter le principe du moindre privilège : ne donnez au service que les autorisations dont il a besoin. Vous pouvez le faire efficacement en déterminant de quelles autorisations le service a besoin et en utilisant des clés de condition globales AWS et des clés de condition AWS KMS pour affiner les autorisations.

Pour trouver les autorisations requises par le service sur une clé gérée par le client, consultez la documentation sur le chiffrement du service. Par exemple, pour connaître les autorisations requises par Amazon Elastic Block Store (Amazon EBS), consultez Autorisations pour les utilisateurs IAM dans le guide de l'utilisateur Amazon EC2 pour les instances Linux et dans le guide de l'utilisateur Amazon EC2 pour les instances Windows. Pour connaître les autorisations requises par Secrets Manager, consultez Autorisation de l'utilisation de la clé KMS dans le guide de l'utilisateur AWS Secrets Manager.

Implémentation des autorisations avec le moindre privilégié

Lorsque vous accordez une autorisation de service AWS pour utiliser une clé KMS, assurez-vous que l'autorisation est valide uniquement pour les ressources auxquelles le service doit accéder en votre nom. Cette stratégie de moindre privilège permet d'empêcher l'utilisation non autorisée d'une clé KMS lorsque les demandes sont transmises entre les services AWS.

Pour implémenter une stratégie de moindre privilège, nous vous recommandons d'utiliser des clés de condition de contexte de chiffrement AWS KMS et des clés de condition ARN source ou de compte source globales.

Utilisation des clés de condition de contexte de chiffrement

Le moyen le plus efficace d'implémenter des autorisations de moindre privilège lors de l'utilisation des ressources AWS KMS consiste à inclure les clés de condition kms:EncryptionContext:context-key ou

kms:EncryptionContextKeys dans la politique qui permet aux principaux d'appeler les opérations de chiffrement AWS KMS. Ces clés de condition sont particulièrement efficaces parce qu'elles associent l'autorisation au contexte de chiffrement qui est lié au texte chiffré lorsque la ressource est chiffrée.

Utilisez les clés de conditions de contexte de chiffrement uniquement lorsque l'action indiquée dans la déclaration de politique est CreateGrantune opération cryptographique AWS KMS symétrique qui prend un `EncryptionContext` paramètre, telle que les opérations telles que GenerateDataKeyou Decrypt. (Pour obtenir la liste des opérations prises en charge, consultez kms:EncryptionContext:context-key ou kms:EncryptionContextKeys.) Si vous utilisez ces clés de condition pour autoriser d'autres opérations, par exemple DescribeKey, l'autorisation sera refusée.

Définissez la valeur sur le contexte de chiffrement utilisé par le service lorsqu'il chiffre la ressource. Ces informations sont généralement disponibles dans le chapitre Sécurité de la documentation du service. Par exemple, le contexte de chiffrement pour Proton AWS identifie la ressource Proton AWS et son modèle associé. Le contexte de chiffrement AWS Secrets Manager identifie le secret et sa version. Le contexte de chiffrement pour Amazon Location identifie le dispositif de suivi ou la collection.

L'exemple suivant d'instruction de politique de clé permet à Amazon Location Service de créer des octrois pour le compte des utilisateurs autorisés. Cette déclaration de politique limite l'autorisation en utilisant les touches kms : CallerAccount, kms : et `kms:EncryptionContext:`*`context-key`* condition pour lier l'autorisation à une ressource de suivi particulière. ViaService

```
{
  "Sid": "Allow Amazon Location to create grants on behalf of authorized users",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/LocationTeam"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "geo.us-west-2.amazonaws.com",
      "kms:CallerAccount": "111122223333",
      "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:tracker/
SAMPLE-Tracker"
    }
  }
}
```

Utilisation des clés de condition **`aws:SourceArn`** ou **`aws:SourceAccount`**

Lorsque le principal dans une instruction de politique de clé est un mandataire du service AWS, nous vous recommandons vivement d'utiliser les clés de condition globales aws:SourceArn ou aws:SourceAccount, en plus de la clé de condition `kms:EncryptionContext:`*`context-key`*. Les valeurs ARN et de compte sont incluses dans le contexte d'autorisation uniquement lorsque AWS KMS reçoit une demande d'un autre service AWS. Cette combinaison de conditions implémente des autorisations de moindre privilège et évite l'éventualité pour un programme d'être manipulé par un autre pour obtenir un accès. Les principaux de service ne sont généralement pas utilisés comme principaux dans une politique de clé, mais certains services AWS, tels que AWS CloudTrail, l'imposent.

Pour utiliser les clés de condition globales `aws:SourceArn` ou `aws:SourceAccount`, définissez comme valeur l'Amazon Resource Name (ARN) ou le compte de la ressource à chiffrer. Par exemple, dans une instruction de politique de clé qui autorise AWS CloudTrail à chiffrer un journal d'activité, définissez l'ARN de ce dernier comme valeur de `aws:SourceArn`. Dans la mesure du possible, utilisez `aws:SourceArn`, qui est plus spécifique. Définissez comme valeur l'ARN ou un modèle d'ARN avec des caractères génériques. Si vous ne connaissez pas l'ARN de la ressource, utilisez `aws:SourceAccount` à la place.

> **Note**
>
> Si un ARN de ressources inclut des caractères non autorisés dans une politique de clé AWS KMS, vous ne pouvez pas utiliser cet ARN de ressource dans la valeur de la clé de condition `aws:SourceArn`. Utilisez à la place la clé de condition `aws:SourceAccount`. Pour plus d'informations sur les règles de document de politique de clé, voir Format de politique de clé.

Dans l'exemple de politique de clé suivant, le principal qui obtient les autorisations est `cloudtrail.amazonaws.com`, le principal du service AWS CloudTrail. Pour implémenter le moindre privilège, cette politique utilise les clés de condition `aws:SourceArn` et `kms:EncryptionContext:`*`context-key`*. La déclaration de politique CloudTrail permet d'utiliser la clé KMS pour générer la clé de données utilisée pour chiffrer une trace. Les conditions `aws:SourceArn` et `kms:EncryptionContext:`*`context-key`* sont évaluées indépendamment. Toute demande d'utilisation de la clé KMS pour l'opération spécifiée doit répondre aux deux conditions.

Pour restreindre l'autorisation du service au journal d'activité `finance` dans l'exemple de compte (111122223333) et la région `us-west-2`, cette instruction de politique affecte à la condition de clé `aws:SourceArn` l'ARN d'un journal d'activité donné. L'instruction de condition utilise l'ArnEqualsopérateur pour garantir que chaque élément de l'ARN est évalué indépendamment lors de la correspondance. L'exemple utilise également la clé de condition `kms:EncryptionContext:`*`context-key`* pour limiter l'autorisation aux journaux d'activité dans un compte et une région particuliers.

Avant d'utiliser cette politique de clé, remplacez l'exemple d'ID de compte, de région et de nom de journal d'activité par des valeurs valides de votre compte.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow CloudTrail to encrypt logs",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "kms:GenerateDataKey",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cloudtrail:us-west-2:111122223333:trail/finance"
          ]
        },
        "StringLike": {
          "kms:EncryptionContext:aws:cloudtrail:arn": [
            "arn:aws:cloudtrail:*:111122223333:trail/*"
          ]
        }
      }
    }
  ]
}
```

Utilisation des politiques IAM avec AWS KMS

Vous pouvez utiliser des politiques IAM, ainsi que des politiques de clé, des octrois et des politiques de points de terminaison d'un VPC pour contrôler l'accès à vos AWS KMS keys dans AWS KMS.

> **Note**
>
> Pour utiliser une politique IAM afin de contrôler l'accès à une clé KMS, la politique de clé de la clé KMS doit donner au compte l'autorisation d'utiliser des politiques IAM. Plus précisément, la politique de clé doit inclure l'instruction de politique qui autorise les politiques IAM.
>
> Cette section explique comment utiliser des politiques IAM pour contrôler l'accès aux opérations AWS KMS. Pour plus d'informations sur IAM, veuillez consulter le Guide de l'utilisateur IAM.

Toutes les clés KMS doivent avoir une politique de clé. Les politiques IAM sont facultatives. Pour utiliser une politique IAM afin de contrôler l'accès à une clé KMS, la politique de clé de la clé KMS doit donner au compte l'autorisation d'utiliser des politiques IAM. Plus précisément, la politique de clé doit inclure l'instruction de politique qui autorise les politiques IAM.

Les politiques IAM peuvent contrôler l'accès à n'importe quelle opération AWS KMS. Contrairement aux politiques de clé, les politiques IAM peuvent contrôler l'accès à plusieurs clés KMS et fournir des autorisations pour les opérations de plusieurs services AWS connexes. Mais les politiques IAM sont particulièrement utiles pour contrôler l'accès aux opérations, par exemple celles CreateKeyqui ne peuvent pas être contrôlées par une politique clé car elles n'impliquent aucune clé KMS en particulier.

Si vous accédez à AWS KMS via un point de terminaison Amazon Virtual Private Cloud (Amazon VPC), vous pouvez également utiliser une politique de point de terminaison d'un VPC pour limiter l'accès à vos ressources AWS KMS lors de l'utilisation du point de terminaison. Par exemple, lorsque vous utilisez le point de terminaison d'un VPC, vous pouvez uniquement autoriser les principaux de votre Compte AWS à accéder à vos clés gérées par le client. Pour plus de détails, veuillez consulter Contrôle de l'accès à votre point de terminaison d'un VPC.

Pour obtenir de l'aide sur la rédaction et la mise en forme d'un document de politique JSON, veuillez consulter Référence de politique JSON IAM dans le Guide de l'utilisateur IAM.

Rubriques

- Présentation des politiques IAM

- Bonnes pratiques pour les politiques IAM
- Spécification de clés KMS dans les instructions de politique IAM
- Autorisations requises pour utiliser la console AWS KMS
- Politique gérée AWS pour les utilisateurs expérimentés
- Exemples de politique IAM

Présentation des politiques IAM

Vous pouvez utiliser les politiques IAM comme suit :

- Attachez une politique d'autorisations à un rôle pour accorder des autorisations de fédération ou entre comptes - Vous pouvez attacher une politique IAM à un rôle IAM pour activer la fédération d'identité, autoriser les autorisations entre comptes ou accorder des autorisations aux applications qui s'exécutent sur les instances EC2. Pour plus d'informations sur les différents cas d'utilisation pour les rôles IAM, veuillez consulter Rôles IAM dans le Guide de l'utilisateur IAM.
- Attribuez une politique d'autorisations à un utilisateur ou à un groupe - Vous pouvez attribuer à un utilisateur ou à un groupe d'utilisateurs une politique qui les autorise à appeler des opérations AWS KMS. Toutefois, chaque fois que possible, les bonnes pratiques IAM recommandent d'utiliser des identités dotées d'informations d'identification temporaires, comme des rôles IAM.

L'exemple suivant montre une politique IAM avec des autorisations AWS KMS. Cette politique autorise les identités IAM auxquelles elle est attachée à répertorier toutes les clés KMS et leurs alias.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

Comme toutes les politiques IAM, elle n'a pas d'élément `Principal`. Lorsque vous attribuez une politique IAM à une identité IAM, cette identité bénéficie des autorisations spécifiées dans la politique.

Pour visualiser un tableau répertoriant toutes les actions d'API AWS KMS, ainsi que les ressources auxquelles elles s'appliquent, consultez la Référence des autorisations .

Bonnes pratiques pour les politiques IAM

La sécurisation de l'accès aux AWS KMS keys est essentielle à la sécurité de toutes vos ressources AWS. Les clés KMS sont utilisées pour protéger la plupart des ressources les plus sensibles de votre Compte AWS. Prenez le temps de concevoir les politiques de clé, les politiques IAM, les octrois et les politiques de point de terminaison d'un VPC qui contrôlent l'accès à vos clés KMS.

Dans les instructions de politique IAM qui contrôlent l'accès aux clés KMS, utilisez le principe du moindre privilège. N'accordez aux principaux IAM que les autorisations dont ils ont besoin pour les clés KMS qu'ils doivent utiliser ou gérer.

Les bonnes pratiques suivantes s'appliquent aux politiques IAM qui contrôlent l'accès aux alias et aux clés AWS KMS. Pour obtenir des conseils généraux sur les bonnes pratiques en matière de politique IAM, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

Utilisation de politiques de clé

Dans la mesure du possible, fournissez des autorisations dans les politiques de clé qui affectent une clé KMS, plutôt que dans une politique IAM qui peut s'appliquer à de nombreuses clés KMS, y compris celles d'autres Comptes AWS. Cela est particulièrement important pour les autorisations sensibles telles que kms : PutKeyPolicy et kms : ScheduleKeyDeletion mais également pour les opérations cryptographiques qui déterminent la manière dont vos données sont protégées.

Limiter CreateKey l'autorisation

Donnez l'autorisation de créer des clés (kms : CreateKey) uniquement aux principaux qui en ont besoin. Les principaux qui créent une clé KMS définissent également sa politique de clé, afin qu'ils puissent se donner, ainsi qu'à d'autres, l'autorisation d'utiliser et de gérer les clés KMS qu'ils créent. Lorsque vous accordez cette autorisation, envisagez de la limiter en utilisant les conditions de politique. Par exemple, vous pouvez utiliser la KeySpec condition kms : pour limiter l'autorisation aux clés KMS de chiffrement symétriques.

Spécifier des clés KMS dans une politique IAM

La bonne pratique consiste à spécifier l'ARN de clé de chaque clé KMS à laquelle l'autorisation s'applique dans l'élément `Resource` de l'instruction de politique. Cette pratique limite

l'autorisation aux clés KMS requises par le principal. Par exemple, cet élément `Resource` ne répertorie que les clés KMS que le principal doit utiliser.

```
"Resource": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
]
```

Lorsque la spécification de clés KMS n'est pas pratique, utilisez une valeur `Resource` qui limite l'accès aux clés KMS dans un Compte AWS et une région de confiance, telle que `arn:aws:kms:`*`region`*`:`*`account`*`:key/*`. Vous pouvez aussi limiter l'accès aux clés KMS dans toutes les régions (*) d'un Compte AWS de confiance, comme `arn:aws:kms:*:`*`account`*`:key/*`.

Vous ne pouvez pas utiliser d'ID de clé, de nom d'alias ou d'ARN d'alias pour représenter une clé KMS dans le champ `Resource` d'une politique IAM. Si vous spécifiez un ARN d'alias, la politique s'applique à l'alias et non à la clé KMS. Pour plus d'informations sur les politiques IAM d'alias, veuillez consulter Contrôle de l'accès aux alias.

Évitez « Resource » : « * » dans une politique IAM.

Utilisez judicieusement les caractères génériques (*). Dans une politique de clé, le caractère générique de l'élément `Resource` représente la clé KMS à laquelle la politique de clé est attachée. Mais dans une politique IAM, un caractère générique seul dans l'élément `Resource` (`"Resource": "*"`) applique les autorisations à toutes les clés KMS dans tous les Comptes AWS que le compte du principal a l'autorisation d'utiliser. Cela pourrait inclure les clés KMS dans d'autres Comptes AWS, ainsi que les clés KMS dans le compte du principal.

Par exemple, pour utiliser une clé KMS dans un autre Compte AWS, un principal a besoin de l'autorisation de la politique de clé de la clé KMS dans le compte externe et d'une politique IAM dans son propre compte. Supposons qu'un compte arbitraire ait donné à votre Compte AWS l'autorisation kms:Decrypt sur leurs clés KMS. Si c'est le cas, une politique IAM de votre compte qui donne à un rôle l'autorisation `kms:Decrypt` sur toutes les clés KMS (`"Resource": "*"`) satisferait à la partie IAM de l'exigence. Par conséquent, les principaux qui peuvent endosser ce rôle peuvent désormais déchiffrer les textes chiffrés à l'aide de la clé KMS du compte non approuvé. Les entrées relatives à leurs opérations apparaissent dans les CloudTrail journaux des deux comptes.

Évitez notamment d'utiliser `"Resource": "*"` dans une instruction de politique qui autorise les opérations d'API suivantes. Ces opérations peuvent être appelées sur les clés KMS dans d'autres Comptes AWS.

- DescribeKey
- GetKeyRotationStatus
- Opérations cryptographiques (chiffrer, déchiffrer,,,, GenerateDataKey, GenerateDataKeyPair,, GenerateDataKeyWithoutPlaintext, GenerateDataKeyPairWithoutPlaintext, signer GetPublicKeyReEncrypt, vérifier)
- CreateGrant, ListGrants, ListRetirableGrants, RetireGrant, RevokeGrant

Quand utiliser « Resource » : « * »

Dans une politique IAM, utilisez un caractère générique dans l'élément `Resource` uniquement pour les autorisations qui le requièrent. Seules les autorisations suivantes nécessitent l'élément `"Resource": "*"`.

- km : CreateKey
- km : GenerateRandom
- km : ListAliases
- km : ListKeys
- Autorisations pour les magasins de clés personnalisés, tels que kms : CreateCustomKeyStore et kms : ConnectCustomKeyStore.

> **Note**
>
> Les autorisations pour les opérations d'alias (kms : CreateAlias, kms : UpdateAlias, kms : DeleteAlias) doivent être associées à l'alias et à la clé KMS. Vous pouvez utiliser `"Resource": "*"` dans une politique IAM pour représenter les alias et les clés KMS, ou spécifier les alias et les clés KMS dans l'élément `Resource`. Pour obtenir des exemples, consultez Contrôle de l'accès aux alias.

Les exemples de cette rubrique fournissent plus d'informations et de conseils sur la conception de politiques IAM pour les clés KMS. Pour des conseils sur les bonnes pratiques AWS KMS, veuillez consulter le document Bonnes pratiques AWS Key Management Service (PDF). Pour connaître

les bonnes pratiques IAM pour toutes les ressources AWS, veuillez consulter Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Spécification de clés KMS dans les instructions de politique IAM

Vous pouvez utiliser une politique IAM pour permettre à un principal d'utiliser ou de gérer des clés KMS. Les clés KMS sont spécifiées dans l'élément `Resource` de l'instruction de politique.

- Pour spécifier une clé KMS dans une instruction de politique IAM, vous devez utiliser son ARN de clé. Vous ne pouvez pas utiliser un ID de clé, un nom d'alias ou un ARN d'alias pour identifier une clé KMS dans une instruction de politique IAM.

  Par exemple : « `Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` »

  Pour contrôler l'accès à une clé KMS en fonction de ses alias, utilisez les clés de ResourceAliases condition kms : RequestAlias ou kms :. Pour plus de détails, consultez ABAC pour AWS KMS.

  Utilisez un alias ARN comme ressource uniquement dans une déclaration de politique qui contrôle l'accès aux opérations d'alias, telles que CreateAliasUpdateAlias, ou DeleteAlias. Pour plus de détails, veuillez consulter Contrôle de l'accès aux alias.
- Pour spécifier plusieurs clés KMS dans le compte et la région, utilisez des caractères génériques (*) dans les positions Region (Région) ou Resource ID (ID de ressource) de l'ARN clé.

  Par exemple, pour spécifier toutes les clés KMS dans la région USA Ouest (Oregon) d'un compte, utilisez « `Resource": "arn:aws:kms:us-west-2:111122223333:key/*` ». Pour spécifier toutes les clés KMS dans toutes les régions du compte, utilisez « `Resource": "arn:aws:kms:*:111122223333:key/*` ».
- Pour représenter toutes les clés KMS, utilisez un caractère générique seul (`"*"`). Utilisez ce format pour les opérations qui n'utilisent aucune clé KMS particulière CreateKey, à savoir GenerateRandomListAliases, et ListKeys.

Lorsque vous rédigez vos instructions de politique, il s'agit d'une bonne pratique pour spécifier uniquement les clés KMS que le principal doit utiliser, plutôt que de leur donner accès à toutes les clés KMS.

Par exemple, la déclaration de politique IAM suivante permet au principal d'appeler les DescribeKeyopérations de déchiffrement uniquement sur les clés KMS répertoriées dans

l'`Resource`élément de la déclaration de politique. GenerateDataKey La spécification des clés KMS par ARN de clé, qui est une bonne pratique, garantit que les autorisations sont limitées uniquement aux clés KMS spécifiées.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

Pour appliquer l'autorisation à toutes les clés KMS d'un Compte AWS de confiance spécifique, vous pouvez utiliser des caractères génériques (*) dans les positions Region (Région) et key ID (ID de clé). Par exemple, l'instruction de politique suivante permet au principal d'appeler les opérations spécifiées sur toutes les clés KMS dans deux exemples de comptes de confiance.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyPair"
    ],
    "Resource": [
      "arn:aws:kms:*:111122223333:key/*",
      "arn:aws:kms:*:444455556666:key/*"
    ]
  }
}
```

Vous pouvez également utiliser un caractère générique (`"*"`) seul dans l'élément `Resource`. Comme il permet l'accès à toutes les clés KMS que le compte a l'autorisation d'utiliser, il est recommandé principalement pour les opérations sans clé KMS particulière et pour les instructions `Deny`. Vous pouvez également l'utiliser dans des instructions de politique qui autorisent uniquement des opérations moins sensibles en lecture seule. Pour déterminer si une opération AWS KMS implique une clé KMS particulière, recherchez la valeur KMS key (clé KMS) dans la colonne Resources (Ressources) de la table dans the section called “Référence des autorisations ”.

Par exemple, l'instruction de politique suivante utilise un effet `Deny` pour interdire aux principaux d'utiliser les opérations spécifiées sur une clé KMS. Elle utilise un caractère générique dans l'élément `Resource` pour représenter toutes les clés KMS.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:CreateKey",
      "kms:PutKeyPolicy",
      "kms:CreateGrant",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

L'instruction de politique suivante utilise un caractère générique seul pour représenter toutes les clés KMS. Mais il n'autorise que les opérations moins sensibles en lecture seule et les opérations qui ne s'appliquent pas à une clé KMS particulière.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:CreateKey",
      "kms:ListKeys",
      "kms:ListAliases",
      "kms:ListResourceTags"
    ],
    "Resource": "*"
```

```
  }
}
```

Autorisations requises pour utiliser la console AWS KMS

Pour utiliser la console AWS KMS, les utilisateurs doivent disposer d'un ensemble minimal d'autorisations qui leur permet de travailler avec les ressources AWS KMS dans leur Compte AWS. Outre ces autorisations AWS KMS, les utilisateurs doivent également être autorisés à répertorier les utilisateurs et rôles IAM. Si vous créez une politique IAM plus restrictive que les autorisations minimales requises, la console AWS KMS ne fonctionnera pas comme prévu pour les utilisateurs dotés de cette politique IAM.

Pour connaître les autorisations minimales requises pour accorder à un utilisateur l'accès en lecture seule à la console AWS KMS, consultez Autoriser un utilisateur à afficher les clés KMS dans la console AWS KMS.

Pour permettre aux utilisateurs d'utiliser la AWS KMS console pour créer et gérer des clés KMS, associez la politique AWSKeyManagementServicePowerUsergérée à l'utilisateur, comme décrit dans la section suivante.

Vous n'avez pas besoin d'accorder les autorisations minimales d'utilisation de la console pour les utilisateurs qui utilisent l'API AWS KMS via les kits SDK AWS, l'AWS Command Line Interface ou AWS Tools for PowerShell. Cependant, vous devez accorder à ces utilisateurs l'autorisation d'utiliser l'API. Pour plus d’informations, consultez Référence des autorisations .

Politique gérée AWS pour les utilisateurs expérimentés

Vous pouvez utiliser une politique gérée par `AWSKeyManagementServicePowerUser` pour accorder aux principaux IAM de votre compte, les autorisations d'un utilisateur avec pouvoir. Les utilisateurs expérimentés peuvent créer des clés KMS, utiliser et gérer les clés KMS qu'ils créent et afficher toutes les clés KMS et les identités IAM. Les principals qui ont la politique gérée `AWSKeyManagementServicePowerUser` peuvent également obtenir des autorisations d'autres sources, notamment des politiques de clé, d'autres politiques IAM et des octrois.

`AWSKeyManagementServicePowerUser` est une politique IAM gérée par AWS. Pour plus d’informations sur les politiques gérées par AWS, consultez Politiques gérées par AWS dans le Guide de l’utilisateur IAM.

> **Note**
>
> Les autorisations de cette politique qui sont spécifiques à une clé KMS, telles que `kms:TagResource` et `kms:GetKeyRotationStatus`, ne sont effectives que lorsque la politique de clé pour cette clé KMS autorise explicitement le Compte AWS à utiliser des politiques IAM pour contrôler l'accès à la clé. Pour déterminer si une autorisation est spécifique à une clé KMS, consultez AWS KMS autorisations et recherchez une valeur de Clé KMS dans la colonne Ressources.
>
> Cette politique accorde à l'utilisateur expérimenté les autorisations sur n'importe quelle clé KMS avec une politique de clé qui permet l'opération. Pour les autorisations entre comptes, telles que `kms:DescribeKey` et `kms:ListGrants`, cela peut inclure des clés KMS dans des Comptes AWS non approuvés. Pour plus d'informations, consultez Bonnes pratiques pour les politiques IAM et Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS. Pour déterminer si une autorisation est valide sur les clés KMS dans d'autres comptes, consultez AWS KMS autorisations et recherchez la valeur Oui dans la colonne Utilisation inter-comptes.
>
> Pour permettre aux principaux d'accéder à la AWS KMS console sans erreur, ils ont besoin de la balise : GetResources permission, qui n'est pas incluse dans la `AWSKeyManagementServicePowerUser` politique. Vous pouvez accorder cette autorisation dans une politique IAM distincte.

La stratégie IAM gérée par AWSKeyManagementServicePower doit inclure les autorisations suivantes.

- Autorise les principals à créer des clés KMS. Comme ce processus comprend la définition de la politique de clé, les utilisateurs avancés peuvent s'accorder à eux-mêmes et à accorder à d'autres l'autorisation d'utiliser et de gérer les clés KMS qu'ils créent.
- Autorise les principals à créer et supprimer des alias et des balises sur toutes les clés KMS. La modification d'une balise ou d'un alias peut autoriser ou interdire l'utilisation et la gestion de la clé KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS.
- Permet aux principals d'obtenir des informations détaillées sur toutes les clés KMS, y compris leur ARN de clé, leur configuration de chiffrement, leur politique de clé, leurs alias, leurs balises et leur statut de rotation.
- Permet aux principals de répertorier les utilisateurs, les groupes et les rôles IAM.

- Cette politique n'autorise pas les principals à utiliser ou à gérer des clés KMS qu'ils n'ont pas créées. Cependant, ils peuvent modifier les alias et les balises sur toutes les clés KMS, ce qui peut leur autoriser ou leur refuser l'autorisation d'utiliser ou de gérer une clé KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateAlias",
                "kms:CreateKey",
                "kms:DeleteAlias",
                "kms:Describe*",
                "kms:GenerateRandom",
                "kms:Get*",
                "kms:List*",
                "kms:TagResource",
                "kms:UntagResource",
                "iam:ListGroups",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemples de politique IAM

Dans cette section, vous trouverez des exemples de politiques IAM qui accordent des autorisations pour diverses actions AWS KMS.

> **Important**
>
> Certaines des autorisations figurant dans les politiques suivantes sont autorisées uniquement lorsque la politique de clé de la clé KMS les autorise également. Pour plus d'informations, consultez Référence des autorisations .

Pour obtenir de l'aide sur la rédaction et la mise en forme d'un document de politique JSON, veuillez consulter Référence de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples

- Autoriser un utilisateur à afficher les clés KMS dans la console AWS KMS
- Autoriser un utilisateur à créer des clés KMS
- Autoriser un utilisateur à chiffrer et déchiffrer avec n'importe quelle clé KMS dans un Compte AWS spécifique
- Autoriser un utilisateur à chiffrer et déchiffrer avec n'importe quelle clé KMS dans un Compte AWS et une région spécifiques
- Autoriser un utilisateur à chiffrer et déchiffrer des données avec des clés KMS spécifiques
- Empêcher un utilisateur de désactiver et de supprimer des clés KMS

Autoriser un utilisateur à afficher les clés KMS dans la console AWS KMS

La politique IAM suivante accorde aux utilisateurs l'accès en lecture seule à la console AWS KMS. Les utilisateurs disposant de ces autorisations peuvent afficher toutes les clés KMS dans leur Compte AWS, mais ils ne peuvent pas créer ou modifier des clés KMS.

Pour afficher les clés KMS sur les pages des clés gérées par le client Clés gérées par AWSet sur les pages des clés gérées par le clientListKeys, les principaux ont besoin des GetResources autorisations kms : ListAliases, kms : et tag :, même si les clés ne comportent pas de balises ni d'alias. Les autorisations restantes, en particulier kms : DescribeKey, sont requises pour afficher les colonnes facultatives du tableau des clés KMS et les données sur les pages détaillées des clés KMS. Les ListRoles autorisations iam : ListUsers et iam : sont requises pour afficher la politique clé dans l'affichage par défaut sans erreur. Pour consulter les données sur la page des magasins de clés personnalisés et les détails sur les clés KMS dans les magasins de clés personnalisés, les principaux ont également besoin de DescribeCustomKeyStores l'autorisation kms :.

Si vous limitez l'accès de la console d'un utilisateur à des clés KMS particulières, la console affiche une erreur pour chaque clé KMS qui n'est pas visible.

Cette politique inclut deux instructions de politique. L'élément `Resource` dans la première instruction de politique accorde les autorisations spécifiées sur toutes les clés KMS dans toutes les régions du Compte AWS d'exemple. Les utilisateurs de la console n'ont pas besoin d'un accès supplémentaire, car la console AWS KMS affiche uniquement les clés KMS dans le compte du principal. Cela est vrai même s'ils ont l'autorisation d'afficher les clés KMS dans d'autres Comptes AWS. Les autorisations

IAM et AWS KMS restantes nécessitent un élément `"Resource": "*"`, car elles ne s'appliquent pas à une clé KMS particulière.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAccessForAllKMSKeysInAccount",
      "Effect": "Allow",
      "Action": [
        "kms:GetPublicKey",
        "kms:GetKeyRotationStatus",
        "kms:GetKeyPolicy",
        "kms:DescribeKey",
        "kms:ListKeyPolicies",
        "kms:ListResourceTags",
        "tag:GetResources"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "ReadOnlyAccessForOperationsWithNoKMSKey",
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys",
        "kms:ListAliases",
        "iam:ListRoles",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Autoriser un utilisateur à créer des clés KMS

La politique IAM suivante permet à un utilisateur de créer tous les types de clés KMS. La valeur de l'élément `Resource` est * parce que l'opération `CreateKey` n'utilise aucune ressource AWS KMS particulière (clé KMS ou alias).

Pour restreindre l'utilisateur à certains types de clés KMS, utilisez les clés de KeyOrigin condition kms : KeyUsage, kms : et kms :. KeySpec

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "kms:CreateKey",
    "Resource": "*"
  }
}
```

Les principaux qui créent des clés peuvent avoir besoin de certaines autorisations associées.

- kms : PutKeyPolicy — Les principaux `kms:CreateKey` autorisés peuvent définir la politique de clé initiale pour la clé KMS. Cependant, l'`CreateKey`appelant doit disposer de l'PutKeyPolicyautorisation kms :, qui lui permet de modifier la politique des clés KMS, ou il doit spécifier le `BypassPolicyLockoutSafetyCheck` paramètre de`CreateKey`, ce qui n'est pas recommandé. L'appelant `CreateKey` peut obtenir l'autorisation `kms:PutKeyPolicy` pour la clé KMS depuis une politique IAM, ou il peut inclure cette autorisation dans la politique de clé de la clé KMS qu'il crée.
- kms : TagResource — Pour ajouter des balises à la clé KMS pendant l'`CreateKey`opération, l'`CreateKey`appelant doit disposer de l'TagResourceautorisation kms : dans une politique IAM. L'inclusion de cette autorisation dans la politique de clé de la nouvelle clé KMS ne suffit pas. Cependant, si l'appelant `CreateKey` inclut `kms:TagResource` dans la politique de clé initiale, il peut ajouter des balises dans un appel séparé après la création de la clé KMS.
- kms : CreateAlias — Les principaux qui créent une clé KMS dans la AWS KMS console doivent disposer de l'CreateAliasautorisation kms : sur la clé KMS et sur l'alias. (La console effectue deux appels ; un à `CreateKey` et un à `CreateAlias`). Vous devez fournir l'autorisation d'alias dans une politique IAM. Vous pouvez fournir l'autorisation de clé KMS dans une politique de clé ou une politique IAM. Pour plus de détails, veuillez consulter Contrôle de l'accès aux alias.

En plus de `kms:CreateKey`, la politique IAM suivante accorde une autorisation `kms:TagResource` sur toutes les clés KMS dans le Compte AWS et une autorisation `kms:CreateAlias` sur tous les alias du compte. Elle inclut également certaines autorisations utiles en lecture seule qui peuvent être fournies uniquement dans une politique IAM.

Cette politique IAM n'inclut pas l'autorisation `kms:PutKeyPolicy` ou toute autre autorisation pouvant être définie dans une politique de clé. La définition de ces autorisations dans la politique de clé, où elles s'appliquent exclusivement à une clé KMS, est une bonne pratique.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPermissionsForParticularKMSKeys",
      "Effect": "Allow",
      "Action": "kms:TagResource",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPermissionsForParticularAliases",
      "Effect": "Allow",
      "Action": "kms:CreateAlias",
      "Resource": "arn:aws:kms:*:111122223333:alias/*"
    },
    {
      "Sid": "IAMPermissionsForAllKMSKeys",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey",
        "kms:ListKeys",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
}
```

Autoriser un utilisateur à chiffrer et déchiffrer avec n'importe quelle clé KMS dans un Compte AWS spécifique

La politique IAM suivante permet à un utilisateur de chiffrer et de déchiffrer des données avec n'importe quelle clé KMS dans le Compte AWS 111122223333.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
```

```
    "Resource": "arn:aws:kms:*:111122223333:key/*"
  }
}
```

Autoriser un utilisateur à chiffrer et déchiffrer avec n'importe quelle clé KMS dans un Compte AWS et une région spécifiques

La politique IAM suivante permet à un utilisateur de chiffrer et de déchiffrer des données avec n'importe quelle clé KMS dans le Compte AWS `111122223333`, dans la région USA Ouest (Oregon).

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/*"
    ]
  }
}
```

Autoriser un utilisateur à chiffrer et déchiffrer des données avec des clés KMS spécifiques

La politique IAM suivante permet à un utilisateur de chiffrer et de déchiffrer des données avec les deux clés KMS spécifiées dans l'élément `Resource`. Lorsque vous spécifiez une clé KMS dans une instruction de politique IAM, vous devez utiliser l'ARN de clé de la clé KMS.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
      "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

Empêcher un utilisateur de désactiver et de supprimer des clés KMS

La politique IAM suivante empêche un utilisateur de désactiver et de supprimer des clés KMS, même si une autre politique IAM ou une politique de clé accorde ces autorisations. Une politique qui refuse explicitement des autorisations se substitue à toutes les autres politiques, même à celles qui accordent explicitement les mêmes autorisations. Pour plus d'informations, consultez Résolution des problèmes de clé d'accès.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:DisableKey",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

Octrois dans AWS KMS

Un octroi est un instrument de politique qui permet aux principaux AWS d'utiliser des clés KMS dans les opérations de chiffrement. Il peut également leur permettre d'afficher une clé KMS (`DescribeKey`), mais aussi de créer et de gérer des octrois. Lorsque vous autorisez l'accès à une clé KMS, les octrois sont pris en compte avec des politiques de clé et des politiques IAM. Les octrois sont souvent utilisés pour des autorisations temporaires, car vous pouvez en créer un, utiliser ses autorisations et les supprimer sans modifier vos politiques de clé ou IAM.

Les octrois sont couramment utilisés par des services AWS qui s'intègrent à AWS KMS pour chiffrer vos données au repos. Le service crée un octroi au nom d'un utilisateur du compte, utilise ses autorisations et retire l'octroi dès que sa tâche est terminée. Pour plus d'informations sur la manière dont les services AWS utilisent les octrois, reportez-vous à la rubrique Comment les services AWS utilisent AWS KMS ou Chiffrement au repos dans le guide de l'utilisateur ou le guide du développeur du service.

Pour découvrir des exemples de code illustrant la façon d'utiliser les octrois dans plusieurs langages de programmation, veuillez consulter Utilisation d'octrois.

Rubriques

- À propos des octrois
- Concepts d'octroi
- Bonnes pratiques relatives aux octrois AWS KMS
- Création d'octrois
- Gestion des octrois

À propos des octrois

Les octrois sont un mécanisme de contrôle d'accès très souple et utile. Lorsque vous créez un octroi pour une clé KMS, celui-ci permet au principal bénéficiaire d'appeler les opérations d'octroi spécifiées sur la clé KMS, à condition que toutes les conditions spécifiées dans l'octroi soient remplies.

- Chaque octroi permet d'accéder à exactement une clé KMS. Vous pouvez créer un octroi pour une clé KMS dans un autre Compte AWS.
- Un octroi peut autoriser l'accès à une clé KMS, mais pas le lui refuser.
- Chaque octroi a un principal bénéficiaire. Le principal bénéficiaire peut représenter une ou plusieurs identités dans le même Compte AWS que la clé KMS ou dans un autre compte.
- Un octroi peut uniquement permettre des opérations d'octroi. Les opérations d'octroi doivent être prises en charge par la clé KMS de l'octroi. Si vous spécifiez une opération non prise en charge, la CreateGrantdemande échoue avec une `ValidationError` exception.
- Le principal bénéficiaire peut utiliser les autorisations que l'octroi lui donne sans spécifier l'octroi, comme il le ferait si les autorisations provenaient d'une politique de clé ou d'une politique IAM. Toutefois, étant donné que l'API AWS KMS suit un modèle de cohérence à terme, lorsque vous créez, retirez ou révoquez un octroi, il peut y avoir un bref délai avant que la modification ne soit disponible sur AWS KMS. Pour utiliser immédiatement les autorisations dans un octroi, utilisez un jeton d'octroi.
- Un principal autorisé peut supprimer l'octroi (le retirer ou le révoquer). La suppression d'un octroi élimine toutes les autorisations qu'il accorde. Vous n'avez pas besoin de déterminer les politiques à ajouter ou à supprimer pour annuler l'octroi.
- AWS KMS limite le nombre d'octrois sur chaque clé KMS. Pour plus de détails, consultez Octrois par clé KMS : 50 000.

Soyez prudent lorsque vous créez des octrois et lorsque vous autorisez d'autres personnes à en créer. L'autorisation de créer des subventions a des implications en matière de sécurité, tout comme l'PutKeyPolicyautorisation de définir des politiques en termes de kilomètres.

- Les utilisateurs autorisés à créer des octrois pour une clé KMS (`kms:CreateGrant`) peuvent utiliser un octroi pour autoriser les utilisateurs et les rôles, y compris les services AWS, à utiliser la clé KMS. Les principaux peuvent être des identités dans votre propre Compte AWS ou des identités dans un autre compte ou une autre organisation.
- Les octrois ne peuvent autoriser qu'un sous-ensemble d'opérations AWS KMS. Vous pouvez utiliser des octrois pour autoriser les principaux à afficher la clé KMS, à l'utiliser dans les opérations de chiffrement, mais aussi à créer et à retirer des octrois. Pour plus d'informations, veuillez consulter .Opérations d'octroi. Vous pouvez également utiliser des contraintes d'octroi pour limiter les autorisations dans un octroi pour une clé de chiffrement symétrique.
- Les principaux peuvent obtenir l'autorisation de créer des octrois à partir d'une politique de clé ou d'une politique IAM. Les directeurs qui obtiennent `kms:CreateGrant` l'autorisation d'une politique peut créer des subventions pour tout opération d'octroi sur la clé KMS. Ces mandants ne sont pas tenus d'avoir l'autorisation qu'ils accordent sur la clé. Lorsque vous accordez l'autorisation `kms:CreateGrant` dans une politique, vous pouvez utiliser des conditions de politique pour limiter cette autorisation.
- Les principaux peuvent également obtenir l'autorisation de créer des octrois à partir d'un octroi. Ces principaux ne peuvent déléguer que les autorisations qui leur ont été accordées, même s'ils disposent d'autres autorisations provenant d'une politique. Pour plus de détails, consultez Octroi CreateGrant d'autorisation.

Pour obtenir de l'aide sur les concepts liés aux octrois, veuillez consulter la Terminologie relative à l'octroi.

Concepts d'octroi

Pour utiliser efficacement les octrois, vous devez comprendre les termes et les concepts utilisés par AWS KMS.

Contrainte d'octroi

> Condition qui limite les autorisations dans l'octroi. Actuellement, AWS KMS prend en charge les contraintes d'octroi basées sur le contexte de chiffrementdans la demande pour une opération cryptographique. Pour plus de détails, consultez Utilisation des contraintes d'octroi.

ID d'octroi

Identifiant unique d'un octroi pour une clé KMS. Vous pouvez utiliser un identifiant de subvention, ainsi qu'un identifiant clé, pour identifier une autorisation dans une RevokeGrantdemande RetireGrantou.

Opérations d'octroi

Les opérations AWS KMS que vous pouvez autoriser dans un octroi. Si vous spécifiez d'autres opérations, la CreateGrantdemande échoue avec une `ValidationError` exception. Ce sont aussi les opérations qui acceptent un jeton d'octroi. Pour de plus amples informations sur ces autorisations, veuillez consulter la AWS KMS autorisations.

Ces opérations d'octroi représentent effectivement l'autorisation d'utiliser l'opération. Par conséquent, pour l'opération `ReEncrypt`, vous pouvez spécifier `ReEncryptFrom`, `ReEncryptTo` ou les deux `ReEncrypt*`.

Les opérations d'octroi sont les suivantes :

- Opérations cryptographiques
  - Decrypt (Déchiffrer)
  - Encrypt (Chiffrer)
  - GenerateDataKey
  - GenerateDataKeyPair
  - GenerateDataKeyPairWithoutPlaintext
  - GenerateDataKeyWithoutPlaintext
  - GenerateMac
  - ReEncryptFrom
  - ReEncryptTo
  - Sign (Signer)
  - Verify (Vérifier)
  - VerifyMac
- Autres opérations
  - CreateGrant
  - DescribeKey
  - GetPublicKey
  - RetireGrant

Les opérations d'octroi que vous autorisez doivent être prises en charge par la clé KMS de l'octroi. Si vous spécifiez une opération non prise en charge, la CreateGrantdemande échoue avec une `ValidationError` exception. Par exemple, les octrois pour les clés KMS de chiffrement symétrique ne peuvent pas autoriser les opérations Sign (Signer), Verify (Vérifier), `GenerateMac` ou `VerifyMac`. Les octrois pour les clés KMS asymétriques ne peuvent autoriser aucune opération générant des clés de données ou des paires de clés de données.

Jeton d'octroi

L'API AWS KMS suit un modèle de cohérence à terme. Lorsque vous créez un octroi, il se peut qu'il y ait un bref délai avant que le changement ne soit disponible via AWS KMS. La propagation de la modification dans l'ensemble du système prend généralement moins de quelques secondes, mais dans certains cas, cela peut prendre plusieurs minutes. Si vous essayez d'utiliser un octroi avant d'être propagé complètement sur le système, vous pouvez obtenir un message d'accès refusé. Un jeton d'octroi vous permet de faire référence à l'octroi et d'utiliser les autorisations d'octroi immédiatement.

Un jeton d'octroi est une chaîne unique, non secrète, de longueur variable et codée en base64 qui représente un octroi. Vous pouvez utiliser le jeton d'octroi pour identifier l'octroi dans n'importe quelle opération d'octroi. Cependant, comme la valeur du jeton est un résumé de hachage, elle ne révèle aucun détail sur l'octroi.

Un jeton d'octroi est conçu pour être utilisé uniquement jusqu'à ce qu'il se propage complètement sur AWS KMS. Après cela, le principal bénéficiaire peut utiliser l'autorisation dans l'octroi sans fournir de jeton d'octroi ou toute autre preuve de l'octroi. Vous pouvez utiliser un jeton d'octroi à tout moment, mais une fois que l'octroi a atteint une cohérence éventuelle, AWS KMS utilise l'octroi pour déterminer les autorisations, et non le jeton d'octroi.

Par exemple, la commande suivante appelle l'GenerateDataKeyopération. Elle utilise un jeton d'octroi pour représenter l'octroi qui donne à l'appelant (le principal bénéficiaire) l'autorisation d'appeler `GenerateDataKey` sur la clé KMS spécifiée.

```
$ aws kms generate-data-key \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --key-spec AES_256 \
        --grant-token $token
```

Vous pouvez également utiliser le jeton d'octroi pour identifier un octroi dans les opérations qui gèrent les octrois. Par exemple, le directeur sortant peut utiliser un jeton de subvention lors d'un appel à l'RetireGrantopération.

```
$ aws kms retire-grant \
        --grant-token $token
```

`CreateGrant` est la seule opération qui renvoie un jeton d'octroi. Vous ne pouvez pas obtenir de jeton d'autorisation à partir d'une autre AWS KMS opération ou du CloudTrail journal des événements associés à l' CreateGrant opération. Les ListRetirableGrantsopérations ListGrantset renvoient l'ID de subvention, mais pas un jeton de subvention.

Pour plus de détails, consultez Utilisation d'un jeton d'octroi.

Principal bénéficiaire

Identité qui obtient les autorisations spécifiées dans l'octroi. Chaque octroi n'a qu'un seul principal bénéficiaire, mais ce dernier peut représenter plusieurs identités.

Le principal qui accorde les octrois peut être n'importe quel principal AWS, y compris un Compte AWS (racine), un utilisateur IAM, un rôle IAM, un utilisateur ou un rôle fédéré, mais aussi un utilisateur de rôle assumé. Le principal bénéficiaire peut se trouver dans le même compte que la clé KMS ou un autre compte. Toutefois, le principal bénéficiaire ne peut pas être un principal de service, un groupe IAM ou une organisation AWS.

> **Note**
>
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

Retirer (un octroi)

Résilie un octroi. Vous retirez un octroi lorsque vous avez terminé d'utiliser les autorisations.

La révocation et le retrait d'un octroi suppriment l'octroi. Toutefois, le retrait est effectué par un principal spécifié dans l'octroi. La révocation est généralement effectuée par un administrateur de clé. Pour plus de détails, consultez Retrait et révocation d'octrois.

Principal de retrait

Un principal qui peut retirer un octroi. Vous pouvez spécifier un principal de retrait dans un octroi, mais ce n'est pas obligatoire. Le principal de retrait peut être n'importe quel principal AWS, y compris des Comptes AWS, des utilisateurs IAM, des rôles IAM, des utilisateurs fédérés et les utilisateurs de rôle assumé. Le principal de retrait peut se trouver dans le même compte que la clé KMS ou un autre compte.

> **Note**
>
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

En plus du principal de retrait spécifié dans l'octroi, un octroi peut être retiré par le Compte AWS dans lequel l'octroi a été créé. Si l'octroi autorise l'opération `RetireGrant`, le principal bénéficiaire peut retirer l'octroi. En outre, le Compte AWS ou un Compte AWS qui est le principal de retrait peut déléguer l'autorisation de retirer un octroi à un principal IAM dans le même Compte AWS. Pour plus de détails, consultez Retrait et révocation d'octrois.

Révoquer (un octroi)

Résilie un octroi. Vous révoquez un octroi pour refuser activement les autorisations que l'octroi autorise.

La révocation et le retrait d'un octroi suppriment l'octroi. Toutefois, le retrait est effectué par un principal spécifié dans l'octroi. La révocation est généralement effectuée par un administrateur de clé. Pour plus de détails, consultez Retrait et révocation d'octrois.

Cohérence éventuelle (pour les octrois)

L'API AWS KMS suit un modèle de cohérence à terme. Lorsque vous créez, retirez ou révoquez un octroi, il se peut qu'il y ait un bref délai avant que le changement ne soit disponible via AWS KMS. La propagation de la modification dans l'ensemble du système prend généralement moins de quelques secondes, mais dans certains cas, cela peut prendre plusieurs minutes.

Vous pouvez prendre connaissance de ce bref délai si vous obtenez des erreurs inattendues. Par exemple, si vous essayez de gérer un nouvel octroi ou si vous utilisez les autorisations dans un

nouvel octroi avant que l'octroi ne soit connu dans AWS KMS, vous pouvez recevoir une erreur d'accès refusé. Si vous retirez ou révoquez un octroi, le principal bénéficiaire peut toujours utiliser ses autorisations pendant une courte période jusqu'à ce que l'octroi soit complètement supprimé. La stratégie typique consiste à réessayer la demande, et certains kits SDK AWS incluent une logique d'interruption et de relance automatique.

AWS KMS dispose de fonctions pour atténuer ce bref délai.

- Pour utiliser immédiatement les autorisations dans un nouvel octroi, utilisez un jeton d'octroi. Vous pouvez utiliser un jeton d'octroi pour faire référence à un octroi dans n'importe quelle opération d'octroi. Pour obtenir des instructions, veuillez consulter Utilisation d'un jeton d'octroi.
- L'CreateGrantopération possède un `Name` paramètre qui empêche les nouvelles tentatives de créer des autorisations dupliquées.

> **Note**
>
> Les jetons d'octroi remplacent la validité de l'octroi jusqu'à ce que tous les points de terminaison du service aient été mis à jour avec le nouvel état de l'octroi. Dans la plupart des cas, une cohérence éventuelle sera obtenue dans les cinq minutes.

Pour plus d'informations, consultez la rubrique relative à la cohérence à terme AWS KMS.

Bonnes pratiques relatives aux octrois AWS KMS

AWS KMS recommande les bonnes pratiques suivantes lorsqu'il s'agit de créer, d'utiliser et de gérer des octrois.

- Limitez les autorisations de l'octroi aux autorisations requises par le principal bénéficiaire. Utilisez le principe d'accès le moins privilégié.
- Utilisez un principal bénéficiaire spécifique, tel qu'un rôle IAM, et donnez au principal l'autorisation d'utiliser uniquement les opérations API dont il a besoin.
- Utilisez le contexte de chiffrement de contraintes d'octroi pour garantir que les appelants utilisent la clé KMS aux fins prévues. Pour en savoir plus sur l'utilisation du contexte de chiffrement dans une demande de sécurisation de vos données, consultez Comment protéger l'intégrité de vos données chiffrées en utilisant AWS Key Management Service et EncryptionContext dans le blog sur la AWS sécurité.

> **Tip**
>
> Utilisez la contrainte de EncryptionContextEqualsubvention dans la mesure du possible. La contrainte de EncryptionContextSubsetsubvention est plus difficile à utiliser correctement. Si vous devez l'utiliser, lisez attentivement la documentation et testez la contrainte d'octroi pour vous assurer qu'elle fonctionne comme prévu.

- Supprimer les octrois en double. Les octrois en double ont les mêmes ARN de clé, actions d'API, principal bénéficiaire, contexte de chiffrement et nom. Si vous retirez ou révoquez l'octroi initial, mais que vous laissez les doublons, les doublons restants constituent une escalade involontaire de privilèges. Pour éviter de dupliquer les octrois lors de la relance d'une demande `CreateGrant`, utilisez le paramètre `Name`. Pour détecter les autorisations dupliquées, utilisez l'ListGrantsopération. Si vous créez accidentellement un octroi en double, retirez-le ou révoquez-le dès que possible.

  > **Note**
  >
  > Les octrois pour les clés gérées par AWS peuvent ressembler à des doublons, mais ont des principaux bénéficiaires différents.
  > Le champ `GranteePrincipal` de la réponse `ListGrants` contient habituellement le bénéficiaire principal. Toutefois, lorsque le principal bénéficiaire de l'octroi est un service AWS, le champ `GranteePrincipal` contient le principal de service, qui peut représenter plusieurs principaux bénéficiaires.

- N'oubliez pas que les octrois n'expirent pas automatiquement. Retirez ou révoquez l'octroi dès que l'autorisation n'est plus nécessaire. Les octrois qui ne sont pas supprimés peuvent créer un risque de sécurité pour les ressources chiffrées.

Création d'octrois

Avant de créer un octroi, découvrez ses options de personnalisation. Vous pouvez utiliser des contraintes d'octroi pour limiter les autorisations dans l'octroi. En outre, renseignez-vous sur l'autorisation `CreateGrant` d'octroi. Les principaux qui obtiennent l'autorisation de créer des octrois à partir d'un octroi sont limités au niveau des octrois qu'ils peuvent créer.

Rubriques

- Création d'un octroi
- Utilisation des contraintes d'octroi
- Octroi CreateGrant d'autorisation

Création d'un octroi

Pour créer une subvention, appelez l'CreateGrantopération. Spécifiez une clé KMS, un principal bénéficiaire et une liste des opérations d'octroi autorisées. Vous pouvez également désigner un principal de retrait facultatif. Pour personnaliser l'octroi, utilisez des paramètres `Constraints` facultatifs pour définir les contraintes d'octroi.

Lorsque vous créez, retirez ou révoquez un octroi, il peut y avoir un bref délai, généralement de moins de cinq minutes, avant que la modification ne soit disponible sur AWS KMS. Pour plus d'informations, consultez la rubrique relative à la cohérence à terme (pour les octrois).

Par exemple, la commande `CreateGrant` suivante crée un octroi qui permet aux utilisateurs autorisés à assumer le rôle `keyUserRole` d'appeler l'opération Decrypt (Déchiffrer) sur la clé KMS symétrique spécifiée. L'autorisation utilise le paramètre `RetiringPrincipal` pour désigner un principal qui peut retirer l'autorisation. Elle inclut également une contrainte d'autorisation qui l'autorise uniquement lorsque le contexte de chiffrement de la requête inclut `"Department": "IT"`.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

Si votre code relance l'opération `CreateGrant`, ou utilise un kit SDK AWS qui relance automatiquement les demandes, utilisez le paramètre Name (Nom) facultatif pour empêcher la création d'octrois en double. Si AWS KMS obtient une demande `CreateGrant` pour un octroi avec les mêmes propriétés qu'un octroi existant, y compris le nom, il reconnaît la demande comme une nouvelle tentative et ne crée pas d'autre octroi. Vous pouvez utiliser la valeur `Name` pour identifier l'octroi dans n'importe quelle opération AWS KMS.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans le nom de l'octroi. Il peut apparaître en texte brut dans CloudTrail les journaux et autres sorties.

```
$ aws kms create-grant \
    --name IT-1234abcd-keyUserRole-decrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

Pour découvrir des exemples de code illustrant la façon d'utiliser les octrois dans plusieurs langages de programmation, veuillez consulter Utilisation d'octrois.

Utilisation des contraintes d'octroi

Les contraintes d'octroi définissent les conditions des autorisations que l'octroi donne au principal bénéficiaire. Les contraintes d'octroi prennent la place de clés de condition dans une politique de clé ou une politique IAM. Chaque valeur de contrainte d'octroi peut inclure jusqu'à 8 paires de contextes de chiffrement. La valeur de contexte de chiffrement dans chaque contrainte d'octroi ne peut pas dépasser 384 caractères.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

AWS KMS prend en charge deux contraintes d'octroi, `EncryptionContextEquals` et `EncryptionContextSubset`, qui établissent toutes les deux les exigences relatives au contexte de chiffrement dans une demande d'opération de chiffrement.

Les contraintes d'octroi du contexte de chiffrement sont conçues pour être utilisées avec des opérations d'octroi qui ont un paramètre de contexte de chiffrement.

- Les contraintes de contexte de chiffrement ne sont valides que dans un octroi pour une clé KMS de chiffrement symétrique. Les opérations de chiffrement avec d'autres clés KMS ne prennent pas en charge un contexte de chiffrement.
- La contrainte de contexte de chiffrement est ignorée pour les opérations `DescribeKey` et `RetireGrant`. `DescribeKey` et `RetireGrant` n'ont pas de paramètre de contexte de chiffrement, mais vous pouvez inclure ces opérations dans un octroi qui a une contrainte de contexte de chiffrement.
- Vous pouvez utiliser une contrainte de contexte de chiffrement dans un octroi pour l'opération `CreateGrant`. La contrainte de contexte de chiffrement nécessite que tous les octrois créés avec l'autorisation `CreateGrant` aient une contrainte de contexte de chiffrement tout aussi stricte ou plus stricte.

AWS KMS prend en charge les contraintes d'octroi de contexte de chiffrement suivantes.

EncryptionContextEquals

Utilisez `EncryptionContextEquals` pour spécifier le contexte de chiffrement exact pour les demandes autorisées.

`EncryptionContextEquals` exige que les paires de contexte de chiffrement de la requête correspondent exactement, y compris au niveau des minuscules/majuscules, aux paires de contexte de chiffrement de la contrainte d'octroi. Les paires peuvent apparaître dans n'importe quel ordre, mais les clés et valeurs dans chaque paire ne peuvent pas varier.

Par exemple, si l'a contrainte d'octroi `EncryptionContextEquals` exige la paire de contextes de chiffrement `"Department": "IT"`, l'octroi autorise les demandes du type spécifié uniquement lorsque le contexte de chiffrement de la requête est exactement `"Department": "IT"`.

EncryptionContextSubset

Utilisez `EncryptionContextSubset` pour exiger que les demandes incluent des paires de contexte de chiffrement particulières.

`EncryptionContextSubset` exige que la demande inclue toutes les paires de contexte de chiffrement de la contrainte d'octroi (une correspondance exacte sensible à la casse), mais la demande peut avoir des paires de contexte de chiffrement supplémentaires. Les paires peuvent apparaître dans n'importe quel ordre, mais les clés et valeurs dans chaque paire ne peuvent pas varier.

Par exemple, si l'a contrainte d'octroi `EncryptionContextSubset` exige la paire de contextes de chiffrement `Department=IT`, l'octroi autorise les demandes du type spécifié uniquement lorsque le contexte de chiffrement de la requête est `"Department": "IT"` ou inclut `"Department": "IT"`, ainsi que d'autres paires de contexte de chiffrement, telles que `"Department": "IT","Purpose": "Test"`.

Pour spécifier une contrainte de contexte de chiffrement dans une autorisation pour une clé KMS de chiffrement symétrique, utilisez le `Constraints` paramètre dans l'CreateGrantopération. L'octroi créé par cette commande accorde aux utilisateurs qui sont autorisés à assumer le rôle `keyUserRole` l'autorisation d'appeler l'opération Decrypt (Déchiffrer). Toutefois, cette autorisation est effective uniquement lorsque le contexte de chiffrement de la demande `Decrypt` est une paire de contextes de chiffrement `"Department": "IT"`.

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextEquals={Department=IT}
```

L'octroi obtenu ressemble à ce qui suit. Notez que l'autorisation accordée au rôle `keyUserRole` n'est effective que lorsque la demande `Decrypt` utilise la même paire de contextes de chiffrement que celle spécifiée dans la contrainte d'octroi. Pour trouver les autorisations sur une clé KMS, utilisez l'ListGrantsopération.

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "Decrypt"
            ],
            "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
            "Constraints": {
                "EncryptionContextEquals": {
                    "Department": "IT"
```

```
                }
            },
            "CreationDate": 1568565290.0,
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole"
        }
    ]
}
```

Pour satisfaire la contrainte d'octroi `EncryptionContextEquals`, le contexte de chiffrement dans la demande pour l'opération `Decrypt` doit être une paire `"Department": "IT"`. Une demande telle que la suivante émanant du principal bénéficiaire satisferait à la contrainte d'octroi `EncryptionContextEquals`.

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab\
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT
```

Lorsque la contrainte d'octroi est `EncryptionContextSubset`, les paires de contexte de chiffrement de la demande doivent inclure les paires de contexte de chiffrement dans la contrainte d'octroi, mais la demande peut également inclure d'autres paires de contexte de chiffrement. La contrainte d'octroi suivante nécessite que l'une des paires de contexte de chiffrement dans la demande soit `"Deparment": "IT"`.

```
"Constraints": {
   "EncryptionContextSubset": {
       "Department": "IT"
   }
}
```

La demande suivante émanant du principal bénéficiaire satisferait à la fois aux contraintes d'octroi `EncryptionContextEqual` et `EncryptionContextSubset` dans cet exemple.

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --ciphertext-blob fileb://encrypted_msg \
```

```
    --encryption-context Department=IT
```

Toutefois, une demande comme celle qui suit émanant du principal bénéficiaire satisferait à la contrainte d'octroi `EncryptionContextSubset`, mais pas à la contrainte d'octroi `EncryptionContextEquals`.

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT,Purpose=Test
```

Les services AWS utilisent souvent des contraintes de contexte de chiffrement dans les octrois qui leur accordent l'autorisation d'utiliser des clés KMS dans votre Compte AWS. Par exemple, Amazon DynamoDB utilise un octroi comme le suivant pour obtenir l'autorisation d'utiliser la Clé gérée par AWS pour DynamoDB dans votre compte. La contrainte d'octroi `EncryptionContextSubset` de cet octroi rend les autorisations de l'octroi effectives uniquement lorsque le contexte de chiffrement de la demande inclut les paires`"subscriberID": "111122223333"` et `"tableName": "Services"`. Cette contrainte d'octroi signifie que l'octroi autorise DynamoDB à utiliser la clé KMS spécifiée uniquement pour une table particulière de votre Compte AWS.

Pour obtenir ce résultat, exécutez l'ListGrantsopération sur DynamoDB de votre compte. Clé gérée par AWS

```
$ aws kms list-grants --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
    "Grants": [
        {
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Constraints": {
                "EncryptionContextSubset": {
```

```
                    "aws:dynamodb:tableName": "Services",
                    "aws:dynamodb:subscriberId": "111122223333"
                }
            },
            "CreationDate": 1518567315.0,
            "KeyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59"
        }
    ]
}
```

Octroi CreateGrant d'autorisation

Un octroi peut inclure l'autorisation d'appeler l'opération `CreateGrant`. Toutefois, quand un principal bénéficiaire obtient l'autorisation d'appeler `CreateGrant` à partir d'un octroi, plutôt que d'une politique, cette autorisation est limitée.

- Le principal bénéficiaire peut uniquement créer des octrois qui permettent une partie ou la totalité des opérations de l'octroi parent.
- Les contraintes d'octroi dans les octrois qu'elles créent doivent être au moins aussi strictes que celles de l'octroi parent.

Ces limites ne s'appliquent pas aux principaux qui obtiennent l'autorisation `CreateGrant` à partir d'une politique, bien que leurs autorisations puissent être limitées par des conditions de politique.

Par exemple, imaginons un octroi qui autorise le principal bénéficiaire à appeler les opérations `GenerateDataKey`, `Decrypt` et `CreateGrant`. Nous appelons un octroi qui autorise l'autorisation `CreateGrant` d'un octroi parent.

```
# The original grant in a ListGrants response.
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
```

```
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "GenerateDataKey",
                "Decrypt",
                "CreateGrant
            ]
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
        }
    ]
}
```

Le principal bénéficiaire, exampleUser, peut utiliser cette autorisation pour créer un octroi qui inclut n'importe quel sous-ensemble des opérations spécifiées dans l'octroi parent, par exemple `CreateGrant` et `Decrypt`. L'octroi enfant ne peut pas inclure d'autres opérations, comme `ScheduleKeyDeletion` ou `ReEncrypt`.

De plus, les contraintes d'octroi des octrois enfants doivent être aussi restrictives, voire plus, que celles de l'octroi parent. Par exemple, l'octroi enfant peut ajouter des paires dans une contrainte `EncryptionContextSubset` de l'octroi parent, mais ne peut pas les supprimer. L'octroi enfant peut modifier une contrainte `EncryptionContextSubset` en contrainte `EncryptionContextEquals`, mais pas l'inverse.

Par exemple, le principal bénéficiaire peut utiliser l'autorisation `CreateGrant` qu'il a obtenue de l'octroi parent pour créer l'octroi enfant suivant. Les opérations de l'octroi enfant sont un sous-ensemble des opérations de l'octroi parent et les contraintes d'octroi sont plus restrictives.

```
# The child grant in a ListGrants response.
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572249600.0,
```

```
            "GrantId":
 "fedcba9999c1e2e9876abcde6e9d6c9b6a1987650000abcee009abcdef40183f",
            "Operations": [
                "CreateGrant"
                "Decrypt"
            ]
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:user/anotherUser",
            "Constraints": {
IAM best practices discourage the use of IAM users with long-term credentials. Whenever
                      possible, use IAM roles, which provide temporary credentials. For
 details,
                      see Security best practices in IAM in the IAM User Guide.
        "EncryptionContextEquals": {
                    "Department": "IT"
                }
            },
        }
    ]
}
```

Le principal bénéficiaire de l'octroi enfant ,`anotherUser`, peut utiliser son autorisation `CreateGrant` pour créer des octrois. Cependant, les octrois que `anotherUser` crée doit inclure les opérations dans leur octroi parent ou un sous-ensemble, et les contraintes d'octroi doivent être les mêmes ou plus strictes.

Gestion des octrois

Les entités ayant les autorisations requises peuvent afficher, utiliser et supprimer (retirer ou révoquer) des octrois. Pour affiner les autorisations de création et de gestion des octrois, AWS KMS prend en charge plusieurs conditions de politique que vous pouvez utiliser dans les politiques de clé et les politiques IAM.

Rubriques

- Contrôle de l'accès aux octrois
- Affichage d'octrois
- Utilisation d'un jeton d'octroi
- Retrait et révocation d'octrois

Contrôle de l'accès aux octrois

Vous pouvez contrôler l'accès aux opérations qui créent et gèrent des octrois dans les politiques de clés, les politiques IAM et les octrois. Les principaux qui obtiennent l'autorisation `CreateGrant` d'un octroi ont des autorisations d'octroi plus limitées.

Opération API	politique de clé ou politique IAM	Grant (Octroi)
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-
Retirer des octrois	(Limité. Voir Retrait et révocation d'octrois)	✓
RevokeGrant	✓	-

Lorsque vous utilisez une politique de clé ou IAM pour contrôler l'accès aux opérations qui créent et gèrent des octrois, vous pouvez utiliser une ou plusieurs des conditions de politique suivantes pour limiter l'autorisation. AWS KMS prend en charge toutes les clés de condition suivantes associées à l'octroi. Pour plus d'informations et d'exemples, veuillez consulter AWS KMS clés de condition.

km : GrantConstraintType

Permet aux principaux de créer un octroi uniquement lorsque l'octroi inclut la contrainte d'octroispécifiée.

km : GrantIsFor AWSResource

Permet aux principaux d'appeler `CreateGrant`, `ListGrants`, ou `RevokeGrant` seulement lorsque , un service AWS qui est intégré à AWS KMS, envoie la demande au nom du principal.

km : GrantOperations

Autorise les principaux à créer un octroi, mais limite l'octroi aux opérations spécifiées.

km : GranteePrincipal

Autorise les principaux à créer un octroi uniquement pour le principal bénéficiaire spécifié.

km : RetiringPrincipal

Permet aux principaux de créer un octroi uniquement lorsque l'octroi spécifie un principal de retrait.

Affichage d'octrois

Pour consulter la subvention, utilisez l'ListGrantsopération. Vous devez spécifier la clé KMS à laquelle les octrois s'appliquent. Vous pouvez également filtrer la liste des octrois par ID d'octroi ou principal bénéficiaire. Pour obtenir plus d'exemples, consultez Affichage d'un octroi.

Pour consulter toutes les subventions accordées dans la région Compte AWS et dont le capital est retraité en particulier, utilisez ListRetirableGrants. Les réponses comprennent des détails sur chaque octroi.

> **Note**
>
> Le champ `GranteePrincipal` de la réponse `ListGrants` contient habituellement le bénéficiaire principal. Toutefois, lorsque le principal bénéficiaire de l'octroi est un service AWS, le champ `GranteePrincipal` contient le principal de service, qui peut représenter plusieurs principaux bénéficiaires.

Par exemple, la commande suivante répertorie tous les octrois d'une clé KMS.

```
$  aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
```

```
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Operations": [
                "Decrypt"
            ]
        }
    ]
}
```

Utilisation d'un jeton d'octroi

L'API AWS KMS suit un modèle de cohérence à terme. Lorsque vous créez un octroi, il se peut qu'il ne soit pas effectif immédiatement. Il se peut qu'il y ait un bref délai avant que le changement ne soit disponible via AWS KMS. La propagation de la modification dans l'ensemble du système prend généralement moins de quelques secondes, mais dans certains cas, cela peut prendre plusieurs minutes. Une fois que la modification a été entièrement appliquée à l'ensemble du système, le principal bénéficiaire peut utiliser les autorisations dans l'octroi sans spécifier le jeton d'octroi ou une preuve de l'octroi. Cependant, si un octroi est nouveau et qu'il n'est pas encore connu de tout le AWS KMS, la demande peut échouer avec une erreur `AccessDeniedException`.

Pour utiliser immédiatement les autorisations dans un nouvel octroi, utilisez le jeton d'octroi pour l'octroi. Enregistrez le jeton de subvention renvoyé par l'CreateGrantopération. Ensuite, envoyez le jeton d'octroi dans la demande pour l'opération AWS KMS. Vous pouvez envoyer un jeton d'octroi à AWS KMS n'importe quelle opération d'octroi et envoyer plusieurs jetons d'octroi dans la même demande.

L'exemple suivant utilise l'`CreateGrant`opération pour créer une autorisation autorisant les opérations GenerateDataKeyet Decrypt. Elle enregistre le jeton d'octroi que `CreateGrant` renvoie dans la variable `token`. Ensuite, dans un appel à l'opération `GenerateDataKey`, elle utilise le jeton d'octroi dans la variable `token`.

```
# Create a grant; save the grant token
$ token=$(aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/appUser \
    --retiring-principal arn:aws:iam::111122223333:user/acctAdmin \
    --operations GenerateDataKey Decrypt \
    --query GrantToken \
    --output text)
```

```
# Use the grant token in a request
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --grant-tokens $token
```

Les principaux autorisés peuvent également utiliser un jeton d'octroi pour retirer un nouvel octroi avant même que l'octroi ne soit disponible sur l'ensemble de AWS KMS. (L'opération `RevokeGrant` n'accepte pas de jeton d'octroi.) Pour plus de détails, consultez Retrait et révocation d'octrois.

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

Retrait et révocation d'octrois

Pour supprimer un octroi, la retirer ou la révoquer.

Les RevokeGrantopérations RetireGrantet sont très similaires les unes aux autres. Les deux opérations suppriment un octroi, ce qui élimine les autorisations qu'il accorde. La principale différence entre ces opérations est la façon dont elles sont autorisées.

RevokeGrant

Comme la plupart des opérations AWS KMS, l'accès à l'opération `RevokeGrant` est contrôlé par des politiques de clé et des politiques IAM. L'RevokeGrantAPI peut être appelée par n'importe quel principal `kms:RevokeGrant` autorisé. Cette autorisation est incluse dans les autorisations standard accordées aux administrateurs de clé. En règle générale, les administrateurs révoquent un octroi pour refuser les autorisations qu'il accorde.

RetireGrant

L'octroi détermine qui peut la retirer. Cette conception vous permet de contrôler le cycle de vie d'un octroi sans modifier les politiques clé ou les politiques IAM. Généralement, vous retirez un octroi lorsque vous avez terminé d'utiliser ses autorisations.

Un octroi peut être retiré par un principal de retrait spécifié dans l'octroi. Le principal bénéficiaire peut également retirer l'octroi, mais seulement s'il est également un principal de retrait ou si l'octroi comprend l'opération `RetireGrant`. En tant que sauvegarde, le Compte AWS dans lequel l'octroi a été créé peut retirer l'octroi.

Il existe une autorisation `kms:RetireGrant` qui peut être utilisée dans les politiques IAM, mais dont l'utilité est limitée. Les principaux spécifiés dans l'octroi peuvent retirer un octroi sans l'autorisation `kms:RetireGrant`. L'autorisation `kms:RetireGrant` à elle seule ne permet pas aux principaux de retirer un octroi. L'autorisation `kms:RetireGrant` n'est pas efficace dans une politique de clé.

- Pour refuser l'autorisation de retirer un octroi, vous pouvez utiliser une action `Deny` à l'aide de l'autorisation `kms:RetireGrant`.
- Le Compte AWS propriétaire de la clé KMS peut déléguer l'autorisation `kms:RetireGrant` à un principal IAM du compte.
- Si le principal de retrait est un autre Compte AWS, les administrateurs de l'autre compte peuvent utiliser `kms:RetireGrant` pour déléguer l'autorisation de retirer l'octroi à un principal IAM de ce compte.

L'API AWS KMS suit un modèle de cohérence à terme. Lorsque vous créez, retirez ou révoquez un octroi, il se peut qu'il y ait un bref délai avant que le changement ne soit disponible via AWS KMS. La propagation de la modification dans l'ensemble du système prend généralement moins de quelques secondes, mais dans certains cas, cela peut prendre plusieurs minutes. Si vous devez supprimer un nouvel octroi immédiatement, avant qu'il ne soit disponible dans AWS KMS, utilisez un jeton d'octroi afin de retirer l'octroi. Vous ne pouvez pas utiliser un jeton d'octroi pour révoquer un octroi.

Connexion à AWS KMS via un point de terminaison d'un VPC

Vous pouvez vous connecter directement à AWS KMS via un point de terminaison d'interface privé dans votre cloud privé virtuel (VPC). Lorsque vous utilisez un point de terminaison d'un VPC d'interface, la communication entre votre VPC et AWS KMS est gérée au sein du réseau AWS.

AWS KMS prend en charge les points de terminaison Amazon Virtual Private Cloud (Amazon VPC) à technologie AWS PrivateLink. Chaque point de terminaison d'un VPC est représenté par une ou plusieurs interfaces réseau Elastic (ENI) avec des adresses IP privées dans vos sous-réseaux VPC.

Le point de terminaison de VPC d'interface connecte votre VPC directement à AWS KMS sans passerelle Internet, périphérique NAT, connexion VPN ni connexion AWS Direct Connect. Les instances de votre VPC ne requièrent pas d'adresses IP publiques pour communiquer avec AWS KMS.

Régions

AWS KMS prend en charge les points de terminaison d'un VPC et les politiques relatives aux points de terminaison d'un VPC dans toutes les Régions AWS dans lesquelles AWS KMS est pris en charge.

Rubriques

- Considérations relatives aux points de terminaison de VPC AWS KMS
- Création d'un point de terminaison de VPC pour AWS KMS
- Connexion à un point de terminaison de VPC AWS KMS
- Contrôle de l'accès à votre point de terminaison d'un VPC
- Utilisation d'un point de terminaison VPC dans une déclaration de politique
- Journalisation de votre point de terminaison d'un VPC

Considérations relatives aux points de terminaison de VPC AWS KMS

Avant de configurer un point de terminaison d'un VPC d'interface pour AWS KMS, veuillez consulter la rubrique Propriétés et limites des points de terminaison d'interface dans le Guide de l'utilisateur AWS PrivateLink.

La prise en charge AWS KMS d'un point de terminaison de VPC inclut les éléments suivants.

- Vous pouvez utiliser votre point de terminaison d'un VPC pour appeler toutes les opérations d'API AWS KMS à partir de votre VPC.
- Vous pouvez créer un point de terminaison d'un VPC d'interface qui se connecte à un point de terminaison de région AWS KMS ou un point de terminaison FIPS AWS KMS.
- Vous pouvez utiliser les journaux AWS CloudTrail pour auditer votre utilisation des clés KMS via le point de terminaison de VPC. Pour plus de détails, veuillez consulter Journalisation de votre point de terminaison d'un VPC.

Création d'un point de terminaison de VPC pour AWS KMS

Vous pouvez créer un point de terminaison d'un VPC pour AWS KMS à l'aide de la console Amazon VPC ou de l'API Amazon VPC. Pour de plus amples informations, veuillez consulter Créer un point de terminaison d'interface dans le Guide AWS PrivateLink.

- Pour créer un point de terminaison de VPC pour AWS KMS, utilisez le nom de service suivant :

```
com.amazonaws.region.kms
```

Par exemple, dans la région USA Ouest (Oregon) (`us-west-2`), le nom du service serait :

```
com.amazonaws.us-west-2.kms
```

- Pour créer un point de terminaison d'un VPC dans connecté à un point de terminaison FIPS AWS KMS, utilisez le nom de service suivant :

  ```
  com.amazonaws.region.kms-fips
  ```

  Par exemple, dans la région USA Ouest (Oregon) (`us-west-2`), le nom du service serait :

  ```
  com.amazonaws.us-west-2.kms-fips
  ```

Pour faciliter l'utilisation du point de terminaison de VPC, vous pouvez activer un nom DNS privé pour votre point de terminaison d'un VPC. Si vous sélectionnez l'option Enable DNS Name (Activer le nom DNS), le nom d'hôte DNS AWS KMS standard est résolu vers votre point de terminaison d'un VPC. Par exemple, `https://kms.us-west-2.amazonaws.com` serait résolu vers un point de terminaison d'un VPC connecté au nom du service `com.amazonaws.us-west-2.kms`.

Cette option facilite l'utilisation du point de terminaison d'un VPC. L'AWS et les kits SDK AWS CLI utilisent le nom d'hôte DNS AWS KMS standard par défaut. Vous n'avez donc pas besoin de spécifier l'URL du point de terminaison d'un VPC dans les applications et les commandes.

Pour de plus amples informations, veuillez consulter Accès à un service via un point de terminaison d'interface dans le Guide AWS PrivateLink.

Connexion à un point de terminaison de VPC AWS KMS

Vous pouvez vous connecter à AWS KMS via le point de terminaison de VPC à l'aide de d'un kit SDK AWS, de l'AWS CLI ou de AWS Tools for PowerShell. Pour spécifier le point de terminaison VPC, utilisez son nom DNS.

Par exemple, cette commande list-keys utilise le paramètre `endpoint-url` pour indiquer le point de terminaison VPC. Pour utiliser une commande comme celle-ci, remplacez l'exemple d'ID de point de terminaison VPC par celui de votre compte.

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcdf5678c90a-09p7654s-us-
east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Si vous avez activé les noms d'hôte privés lorsque vous avez créé votre point de terminaison VPC, vous n'avez pas besoin de spécifier l'URL de point de terminaison VPC dans vos commandes de CLI ou dans la configuration de l'application. Le nom d'hôte DNS AWS KMS standard est résolu vers votre point de terminaison d'un VPC. L'AWS CLI et les kits SDK utilisent ce nom d'hôte par défaut. Vous pouvez donc commencer à utiliser le point de terminaison d'un VPC pour vous connecter à un point de terminaison régional AWS KMS sans rien changer dans vos scripts et applications.

Pour utiliser des noms d'hôte privés, les attributs `enableDnsHostnames` et `enableDnsSupport` de votre VPC doivent avoir la valeur `true`. Pour définir ces attributs, utilisez l'ModifyVpcAttributeopération. Pour plus d'informations, veuillez consulter Afficher et mettre à jour les attributs DNS pour votre VPC dans le Guide de l'utilisateur Amazon VPC.

Contrôle de l'accès à votre point de terminaison d'un VPC

Pour contrôler l'accès à votre point de terminaison d'un VPC pour AWS KMS, attachez une politique de point de terminaison d'un VPC à votre point de terminaison d'un VPC. La politique de point de terminaison détermine si les principaux peuvent utiliser le point de terminaison d'un VPC pour appeler des opérations AWS KMS sur des ressources AWS KMS.

Vous pouvez créer une politique de point de terminaison VPC lorsque vous créez votre point de terminaison, et vous pouvez modifier la politique de point de terminaison d'un VPC à tout moment. Utilisez la console de gestion VPC ou les opérations CreateVpcEndpointor ModifyVpcEndpoint. Vous pouvez également créer et modifier une politique de point de terminaison d'un VPC en utilisant un modèle AWS CloudFormation. Pour obtenir de l'aide sur l'utilisation de la console de gestion de VPC, veuillez consulter Créer un point de terminaison d'interface et Modification d'un point de terminaison d'interface dans le AWS PrivateLinkGuide .

> **Note**
>
> AWS KMS prend en charge les politiques de point de terminaison d'un VPC à partir de juillet 2020. Les points de terminaison d'un VPC pour AWS KMS qui ont été créés avant cette date ont la politique de point de terminaison d'un VPC par défaut, mais vous pouvez la modifier à tout moment.

Pour obtenir de l'aide sur la rédaction et la mise en forme d'un document de politique JSON, veuillez consulter Référence de politique JSON IAM dans le Guide de l'utilisateur IAM.

Rubriques

- À propos des politiques de point de terminaison d'un VPC
- Politique de point de terminaison d'un VPC par défaut
- Création d'une stratégie de point de terminaison de VPC
- Affichage d'une politique de point de terminaison d'un VPC

À propos des politiques de point de terminaison d'un VPC

Pour une demande AWS KMS qui utilise un point de terminaison d'un VPC pour réussir, le principal nécessite des autorisations de deux sources :

- Une politique de clé, une politique IAM ou un octroi doivent accorder au principal l'autorisation d'appeler l'opération sur la ressource (clé KMS ou alias).
- Une politique de point de terminaison d'un VPC doit accorder au principal l'autorisation d'utiliser le point de terminaison pour effectuer la demande.

Par exemple, une politique de clé peut accorder à un principal l'autorisation d'appeler Decrypt sur une clé KMS particulière. Toutefois, la politique de point de terminaison d'un VPC peut ne pas permettre à ce principal d'appeler `Decrypt` sur cette clé KMS à l'aide du point de terminaison.

Une politique de point de terminaison VPC peut également autoriser un principal à utiliser le point de terminaison pour appeler certaines DisableKeyclés KMS. Mais si le principal ne dispose pas de ces autorisations provenant d'une politique de clé, d'une politique IAM ou d'un octroi, la demande échoue.

Politique de point de terminaison d'un VPC par défaut

Chaque point de terminaison d'un VPC dispose d'une politique de point de terminaison d'un VPC, mais vous n'êtes pas tenu de spécifier la politique. Si vous ne spécifiez pas de politique, la politique de point de terminaison par défaut autorise toutes les opérations effectuées par tous les principaux sur toutes les ressources du point de terminaison.

Cependant, pour les ressources AWS KMS, le principal doit également avoir l'autorisation d'appeler l'opération à partir d'une politique de clé, d'une politique IAM, ou d'un octroi. Par conséquent, en

pratique, la politique par défaut indique que si un principal a l'autorisation d'appeler une opération sur une ressource, il peut également l'appeler à l'aide du point de terminaison.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Pour permettre aux principaux d'utiliser le point de terminaison d'un VPC uniquement pour un sous-ensemble de leurs opérations autorisées, créez ou mettre à jour la politique de point de terminaison d'un VPC.

Création d'une stratégie de point de terminaison de VPC

Une politique de point de terminaison d'un VPC détermine si un principal a l'autorisation d'utiliser le point de terminaison d'un VPC pour effectuer des opérations sur une ressource. Pour les ressources AWS KMS, le principal doit également avoir l'autorisation d'effectuer les opérations à partir d'une politique de clé, d'une politique IAM, ou d'un octroi.

Chaque instruction de politique de point de terminaison d'un VPC nécessite les éléments suivants :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

L'instruction de politique ne spécifie pas le point de terminaison d'un VPC. Au lieu de cela, elle s'applique à tout point de terminaison d'un VPC auquel la politique est attachée. Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Voici un exemple de politique de point de terminaison d'un VPC pour AWS KMS. Lorsqu'elle est attachée à un point de terminaison d'un VPC, cette politique autorise `ExampleUser` à utiliser le point de terminaison d'un VPC pour appeler les opérations spécifiées sur les clés KMS spécifiées. Avant

d'utiliser une politique comme celle-ci, remplacez le principal d'exemple et l'ARN de clé avec des valeurs valides de votre compte.

```
{
   "Statement":[
      {
         "Sid": "AllowDecryptAndView",
         "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
         "Effect":"Allow",
         "Action": [
             "kms:Decrypt",
             "kms:DescribeKey",
             "kms:ListAliases",
             "kms:ListKeys"
          ],
         "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
   ]
}
```

AWS CloudTrail journalise toutes les opérations qui utilisent le point de terminaison d'un VPC. Toutefois, vos CloudTrail journaux n'incluent pas les opérations demandées par les principaux sur d'autres comptes ni les opérations relatives aux clés KMS sur d'autres comptes.

En tant que tel, vous pouvez créer une politique de point de terminaison d'un VPC qui empêche les principaux des comptes externes d'utiliser le point de terminaison d'un VPC pour appeler des opérations AWS KMS sur toutes les clés du compte local.

L'exemple suivant utilise la clé de condition PrincipalAccount globale aws : pour refuser l'accès à tous les principaux pour toutes les opérations sur toutes les clés KMS, sauf si le principal se trouve dans le compte local. Avant d'utiliser une politique comme celle-ci, remplacez l'ID de compte d'exemple par un ID valide.

```
{
  "Statement": [
    {
      "Sid": "AccessForASpecificAccount",
      "Principal": {"AWS": "*"},
      "Action": "kms:*",
      "Effect": "Deny",
      "Resource": "arn:aws:kms:*:111122223333:key/*",
```

```
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": "111122223333"
        }
      }
    }
  ]
}
```

Affichage d'une politique de point de terminaison d'un VPC

Pour consulter la politique de point de terminaison VPC d'un point de terminaison, utilisez la console de gestion du VPC ou l'opération. DescribeVpcEndpoints

La commande AWS CLI suivante obtient la politique pour le point de terminaison, avec l'ID de point de terminaison d'un VPC spécifié.

Avant d'utiliser cette commande, remplacez l'exemple d'ID de point de terminaison d'exemple par un ID valide provenant de votre compte.

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==`vpce-1234abcdf5678c90a`].[PolicyDocument]'
--output text
```

Utilisation d'un point de terminaison VPC dans une déclaration de politique

Vous pouvez contrôler l'accès aux ressources et opérations AWS KMS lorsque la demande provient d'un VPC ou utilise un point de terminaison d'un VPC. Pour ce faire, utilisez l'une des clés de condition globales suivantes dans une politique de clé ou une politique IAM.

- Utilisez la clé de condition `aws:sourceVpce` pour accorder ou restreindre l'accès en fonction du point de terminaison d'un VPC.
- Utilisez la clé de condition `aws:sourceVpc` pour accorder ou restreindre l'accès en fonction du VPC qui héberge le point de terminaison privé.

> **Note**
>
> Soyez prudent lorsque vous créez des politiques de clé et des politiques IAM basées sur votre point de terminaison d'un VPC. Si une instruction de politique nécessite que les demandes proviennent d'un VPC ou d'un point de terminaison d'un VPC spécifique, les

demandes en provenance de services AWS intégrés qui utilisent une ressource AWS KMS en votre nom risquent d'échouer. Pour obtenir de l'aide, veuillez consulter Utilisation de conditions de point de terminaison d'un VPC dans des politiques avec des autorisations AWS KMS.

Par ailleurs, la clé de condition `aws:sourceIP` n'est pas en vigueur lorsque la demande provient d'un point de terminaison d'un VPC Amazon. Pour restreindre les requêtes à un point de terminaison VPC, utilisez les clés de condition `aws:sourceVpce` ou `aws:sourceVpc`. Pour de plus amples informations, veuillez consulter Gestion des identités et des accès pour les points de terminaison d'un VPC et les services de points de terminaison d'un VPC dans le Guide AWS PrivateLink.

Vous pouvez utiliser ces clés de condition globales pour contrôler l'accès aux AWS KMS keys (clés KMS), aux alias et aux opérations de CreateKeyce type qui ne dépendent d'aucune ressource en particulier.

Par exemple, l'exemple de politique de clé suivant autorise un utilisateur à effectuer des opérations de chiffrement à l'aide d'une clé KMS uniquement lorsque la demande provient du point de terminaison d'un VPC spécifié. Lorsqu'un utilisateur adresse une demande à AWS KMS, l'ID de point de terminaison de VPC dans la demande est comparé à la valeur de la clé de condition `aws:sourceVpce` dans la politique. S'il n'y a pas de concordance, la requête est refusée.

Pour utiliser une politique comme celle-ci, remplacez l'espace réservé d'ID Compte AWS et les ID de point de terminaison d'un VPC par des valeurs valides pour votre compte.

```
{
    "Id": "example-key-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM policies",
            "Effect": "Allow",
            "Principal": {"AWS":["111122223333"]},
            "Action": ["kms:*"],
            "Resource": "*"
        },
        {
            "Sid": "Restrict usage to my VPC endpoint",
            "Effect": "Deny",
            "Principal": "*",
```

```
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:sourceVpce": "vpce-1234abcdf5678c90a"
                }
            }
        }

    ]
}
```

Vous pouvez également utiliser la clé de condition `aws:sourceVpc` pour restreindre l'accès à vos clés KMS en fonction du VPC dans lequel réside le point de terminaison d'un VPC.

L'exemple de politique de clé suivant autorise des commandes qui gèrent la clé KMS uniquement lorsqu'ils proviennent de `vpc-12345678`. En outre, il autorise les commandes qui utilisent la clé KMS pour des opérations cryptographiques uniquement lorsqu'elles proviennent de `vpc-2b2b2b2b`. Vous pouvez utiliser politique comme celle-ci si une application est en cours d'exécution dans un VPC, mais que vous utilisez un second VPC isolé pour les fonctions de gestion.

Pour utiliser une politique comme celle-ci, remplacez l'espace réservé d'ID Compte AWS et les ID de point de terminaison d'un VPC par des valeurs valides pour votre compte.

```
{
    "Id": "example-key-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow administrative actions from vpc-12345678",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Create*","kms:Enable*","kms:Put*","kms:Update*",
                "kms:Revoke*","kms:Disable*","kms:Delete*",
                "kms:TagResource", "kms:UntagResource"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-12345678"
                }
            }
        },
        {
            "Sid": "Allow key usage from vpc-2b2b2b2b",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Encrypt","kms:Decrypt","kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
            }
        },
        {
            "Sid": "Allow read actions from everywhere",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Describe*","kms:List*","kms:Get*"
            ],
            "Resource": "*",
        }
    ]
}
```

Journalisation de votre point de terminaison d'un VPC

AWS CloudTrail journalise toutes les opérations qui utilisent le point de terminaison d'un VPC. Lorsqu'une requête adressée à AWS KMS utilise un point de terminaison d'un VPC, l'ID de point de terminaison VPC apparaît dans l'entrée de journal AWS CloudTrail qui enregistre la requête. Vous pouvez utiliser l'ID de point de terminaison pour auditer l'utilisation de votre point de terminaison de VPC AWS KMS.

Toutefois, vos CloudTrail journaux n'incluent pas les opérations demandées par les principaux sur d'autres comptes ni les demandes d'AWS KMSopérations sur les clés KMS et les alias sur d'autres comptes. En outre, pour protéger votre VPC, les demandes qui sont refusées par une politique de point de terminaison d'un VPC, mais qui auraient été autorisées, ne sont pas enregistrés dans AWS CloudTrail.

Par exemple, cet exemple d'entrée de journal enregistre une requête GenerateDataKey qui utilise le point de terminaison d'un VPC. Le champ `vpcEndpointId` apparaît à la fin de l'entrée de journal.


```
{
  "eventVersion":"1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "accountId": "111122223333",
    "userName": "Alice"
  },
  "eventTime":"2018-01-16T05:46:57Z",
  "eventSource":"kms.amazonaws.com",
  "eventName":"GenerateDataKey",
  "awsRegion":"eu-west-1",
  "sourceIPAddress":"172.01.01.001",
  "userAgent":"aws-cli/1.14.23 Python/2.7.12 Linux/4.9.75-25.55.amzn1.x86_64
 botocore/1.8.27",
  "requestParameters":{
    "keyId":"1234abcd-12ab-34cd-56ef-1234567890ab",
    "numberOfBytes":128
  },
  "responseElements":null,
  "requestID":"a9fff0bf-fa80-11e7-a13c-afcabff2f04c",
  "eventID":"77274901-88bc-4e3f-9bb6-acf1c16f6a7c",
  "readOnly":true,
  "resources":[{
    "ARN":"arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId":"111122223333",
    "type":"AWS::KMS::Key"
  }],
  "eventType":"AwsApiCall",
  "recipientAccountId":"111122223333",
  "vpcEndpointId": "vpce-1234abcdf5678c90a"
```

```
}
```

Clés de condition pour AWS KMS

Vous pouvez spécifier des conditions dans les politiques clés et les politiques IAM qui contrôlent l'accès aux AWS KMS ressources. L'Instruction de politique est en vigueur uniquement lorsque les conditions sont vérifiées. Par exemple, il est possible d'appliquer une instruction de politique après une date spécifique. Ou, vous pouvez faire en sorte qu'une instruction de politique contrôle l'accès uniquement lorsqu'une valeur spécifique apparaît dans une demande d'API.

Pour spécifier des conditions, utilisez les clés de condition dans l'élément `Condition` d'une instruction de politique avec des opérateurs de condition IAM. Certaines clés de condition s'appliquent généralement à AWS. D'autres sont spécifiques à AWS KMS.

Les valeurs des clés de condition doivent respecter les règles de caractères et de codage des politiques AWS KMS clés et des politiques IAM. Pour plus d'informations sur les règles de document de politique de clé, voir Format de politique de clé. Pour plus d'informations sur les règles du document de politique IAM, veuillez consulter Exigences relatives aux noms IAM dans le Guide de l'utilisateur IAM.

Rubriques

- AWS clés de condition globales
- AWS KMS clés de condition
- AWS KMS clés de condition pour AWS Nitro Enclaves

AWS clés de condition globales

AWS définit des clés de condition globales, un ensemble de clés de conditions de politique pour tous les AWS services qui utilisent IAM pour le contrôle d'accès. AWS KMS prend en charge toutes les clés de condition globales. Vous pouvez les utiliser dans les politiques AWS KMS clés et les politiques IAM.

Par exemple, vous pouvez utiliser la clé de condition PrincipalArn globale aws : pour autoriser l'accès à une AWS KMS key (clé KMS) uniquement lorsque le principal de la demande est représenté par le nom de ressource Amazon (ARN) dans la valeur de la clé de condition. Pour prendre en charge le contrôle d'accès basé sur les attributs (ABAC) dans AWS KMS, vous pouvez utiliser la clé de

condition globale aws :ResourceTag/tag-key dans une politique IAM afin d'autoriser l'accès aux clés KMS avec une balise particulière.

Pour éviter qu'un AWS service ne soit utilisé comme un sous-traitant confus dans une politique où le principal est un directeur deAWS service, vous pouvez utiliser les clés de condition aws:SourceArnou aws:SourceAccountglobales. Pour plus de détails, consultez Utilisation des clés de condition `aws:SourceArn` ou `aws:SourceAccount`.

Pour plus d'informations sur les clés de condition AWS globales, y compris les types de demandes dans lesquels elles sont disponibles, voir Clés contextuelles de conditionsAWS globales dans le guide de l'utilisateur IAM. Pour accéder à des exemples d'utilisation des clés de condition globale dans des politiques IAM, veuillez consulter Contrôle de l'accès aux demandes et Contrôle des clés de balise dans le Guide de l'utilisateur IAM.

Les rubriques suivantes fournissent des conseils spéciaux pour l'utilisation des clés de condition basées sur les adresses IP et les points de terminaison VPC.

Rubriques

- Utilisation de la condition d'adresse IP dans les politiques avec autorisations AWS KMS
- Utilisation de conditions de point de terminaison d'un VPC dans des politiques avec des autorisations AWS KMS

Utilisation de la condition d'adresse IP dans les politiques avec autorisations AWS KMS

Vous pouvez l'utiliser AWS KMS pour protéger vos données dans un AWS service intégré.
Mais soyez prudent lorsque vous spécifiez les opérateurs de condition d'adresse IP ou la clé de `aws:SourceIp` condition dans la même déclaration de politique qui autorise ou refuse l'accès à AWS KMS. Par exemple, la politique décrite dans AWS: Refuse l'accès à la AWS base de l'adresse IP source limite les AWS actions aux demandes provenant de la plage d'adresses IP spécifiée.

Envisagez le scénario suivant :

1. Vous associez une politique telle que celle présentée à l'adresse AWSsuivante : Refuse l'accès à une identité IAM en AWS fonction de l'adresse IP source. Vous définissez la valeur de la clé de condition `aws:SourceIp` sur la plage d'adresses IP de la société de l'utilisateur. D'autres politiques permettant d'utiliser Amazon EBS, Amazon EC2 et AWS KMSsont attribuées à cette identité IAM.

2. L'identité tente d'attacher un volume EBS chiffré à une instance EC2. Cette action échoue avec une erreur d'autorisation bien que l'utilisateur soit autorisé à utiliser l'ensemble des services concernés.

L'étape 2 échoue car la demande AWS KMS de déchiffrement de la clé de données chiffrée du volume provient d'une adresse IP associée à l'infrastructure Amazon EC2. Pour que l'opération aboutisse, la requête doit provenir de l'adresse IP de l'utilisateur d'origine. Étant donné que la politique à l'étape 1 refuse explicitement toutes les demandes provenant d'adresses IP autres que celles spécifiées, Amazon EC2 n'est pas autorisé à déchiffrer la clé de données chiffrée du volume EBS.

Par ailleurs, la clé de condition `aws:sourceIP` n'est pas en vigueur lorsque la demande provient d'un point de terminaison d'un VPC Amazon. Pour restreindre les demandes à un point de terminaison VPC, y compris à un point de terminaison VPCAWS KMS, utilisez les clés de condition `aws:sourceVpce` ou `aws:sourceVpc`. Pour plus d'informations, consultez Points de terminaison d'un VPC - Contrôle de l'utilisation de points de terminaison dans le manuel Guide d'utilisateur Amazon VPC.

Utilisation de conditions de point de terminaison d'un VPC dans des politiques avec des autorisations AWS KMS

AWS KMS prend en charge les points de terminaison Amazon Virtual Private Cloud (Amazon VPC) alimentés par. AWS PrivateLink Vous pouvez utiliser les clés de condition globales suivantes dans les politiques clés et les politiques IAM pour contrôler l'accès aux AWS KMS ressources lorsque la demande provient d'un VPC ou utilise un point de terminaison VPC. Pour plus de détails, veuillez consulter Utilisation d'un point de terminaison VPC dans une déclaration de politique.

- `aws:SourceVpc` limite l'accès aux requêtes à partir du VPC spécifié.
- `aws:SourceVpce` limite l'accès aux requêtes à partir du point de terminaison d'un VPC spécifié.

Si vous utilisez ces clés de condition pour contrôler l'accès aux clés KMS, vous risquez de refuser par inadvertance l'accès aux AWS services utilisés en votre AWS KMS nom.

Prenez soin d'éviter une situation comme dans l'exemple des clés de condition d'adresse IP. Si vous limitez les demandes de clé KMS à un point de terminaison VPC ou VPC, les appels AWS KMS provenant d'un service intégré, tel qu'Amazon S3 ou Amazon EBS, risquent d'échouer. Cela peut se produire même si la requête source provient au final du VPC ou du point de terminaison d'un VPC.

AWS KMS clés de condition

AWS KMS fournit un ensemble de clés de condition que vous pouvez utiliser dans les politiques clés et les politiques IAM. Ces clés de condition sont spécifiques à AWS KMS. Par exemple, vous pouvez utiliser la clé de condition `kms:EncryptionContext:context-key` pour exiger un contexte de chiffrement particulier lorsque vous contrôlez l'accès à une clé KMS de chiffrement symétrique.

Conditions pour une demande d'opération d'API

De nombreuses clés de AWS KMS condition contrôlent l'accès à une clé KMS en fonction de la valeur d'un paramètre dans la demande d' AWS KMS opération. Par exemple, vous pouvez utiliser la clé de KeySpec condition kms : dans une politique IAM pour autoriser l'utilisation de l'CreateKeyopération uniquement lorsque la valeur du `KeySpec` paramètre de la `CreateKey` demande est`RSA_4096`.

Ce type de condition fonctionne même lorsque le paramètre n'apparaît pas dans la demande, par exemple lorsque vous utilisez la valeur par défaut du paramètre. Par exemple, vous pouvez utiliser la clé de KeySpec condition kms : pour permettre aux utilisateurs d'utiliser l'`CreateKey`opération uniquement lorsque la valeur du `KeySpec` paramètre est`SYMMETRIC_DEFAULT`, qui est la valeur par défaut. Cette condition autorise les demandes dont le paramètre `KeySpec` a pour valeur `SYMMETRIC_DEFAULT` et les demandes qui n'ont pas de paramètre `KeySpec`.

Conditions pour les clés KMS utilisées dans les opérations d'API

Certaines clés de AWS KMS condition peuvent contrôler l'accès aux opérations en fonction d'une propriété de la clé KMS utilisée dans l'opération. Par exemple, vous pouvez utiliser la KeyOrigin condition kms : pour autoriser les principaux à appeler GenerateDataKeyune clé KMS uniquement lorsque `Origin` la clé KMS est`AWS_KMS`. Pour savoir si une clé de condition peut être utilisée de cette manière, consultez la description de la clé de condition.

L'opération doit être une opération de ressource de clé KMS, c'est-à-dire une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS  key` dans la colonne `Resources` de l'opération. Si vous utilisez ce type de clé de condition avec une opération qui n'est pas autorisée pour une ressource clé KMS particulière, par exemple ListKeys, l'autorisation n'est pas effective car la condition ne peut jamais être satisfaite. Aucune ressource de clé KMS n'est impliquée dans l'autorisation de l'opération `ListKeys` ni aucune propriété `KeySpec`.

Les rubriques suivantes décrivent chaque clé de AWS KMS condition et incluent des exemples de déclarations de politique illustrant la syntaxe des politiques.

Utilisation d'opérateurs d'ensemble avec des clés de condition

Lorsqu'une condition de stratégie compare deux ensembles de valeurs, tels que le jeu de balises d'une demande et le jeu de balises d'une politique, vous devez indiquer AWS comment comparer les ensembles. IAM définit deux opérateurs d'ensemble, `ForAnyValue` et `ForAllValues`, à cette fin. Utilisez les opérateurs d'ensemble uniquement avec des clés de condition multi-valeurs, qui les nécessitent. N'utilisez pas d'opérateurs d'ensemble avec des clés de condition à valeur unique. Comme toujours, testez vos instructions de politiques de manière approfondie avant de les utiliser au sein d'un environnement de production.

Les clés de condition sont à valeur unique ou multi-valeurs. Pour déterminer si une clé de AWS KMS condition est à valeur unique ou à valeurs multiples, consultez la colonne Type de valeur dans la description de la clé de condition.

- Les clés de condition à valeur unique ont au plus une valeur dans le contexte d'autorisation (la demande ou la ressource). Par exemple, étant donné que chaque appel d'API ne peut provenir que d'une seule Compte AWS, kms : CallerAccount est une clé de condition à valeur unique. N'utilisez pas d'opérateur d'ensemble avec une clé de condition à valeur unique.
- Les clés de condition multi-valeurs ont plusieurs valeurs dans le contexte d'autorisation (la demande ou la ressource). Par exemple, étant donné que chaque clé KMS peut avoir plusieurs alias, kms : ResourceAliases peut avoir plusieurs valeurs. Les clés de condition multi-valeurs nécessitent un opérateur d'ensemble.

Notez que la différence entre les clés de condition à valeur unique et multi-valeurs dépend du nombre de valeurs dans le contexte d'autorisation, et non du nombre de valeurs dans la condition de politique.

> **Warning**
>
> L'utilisation d'un opérateur d'ensemble avec une clé de condition à valeur unique peut créer une instruction de politique trop permissive (ou trop restrictive). Utilisez des opérateurs d'ensemble uniquement avec des clés de condition multi-valeurs.
> Si vous créez ou mettez à jour une politique qui inclut un opérateur `ForAllValues` set avec les clés de contexte ou de `aws:RequestTag/`*`tag-key`* condition kms EncryptionContext : :, AWS KMS renvoie le message d'erreur suivant :
> `OverlyPermissiveCondition: Using the ForAllValues set operator with a single-valued condition key matches requests without the specified`

```
    [encryption context or tag] or with an unspecified [encryption
    context or tag]. To fix, remove ForAllValues.
```

Pour de plus amples informations sur les opérateurs d'ensemble `ForAnyValue` et `ForAllValues`, veuillez consulter Utilisation de plusieurs clés et valeurs dans le Guide de l'utilisateur IAM. Pour plus d'informations sur le risque lié à l'utilisation de l'opérateur `ForAllValues` défini avec une condition à valeur unique, voir Avertissement de sécurité : ForAllValues clé à valeur unique dans le guide de l'utilisateur IAM.

Rubriques

- km : BypassPolicyLockoutSafetyCheck
- km : CallerAccount
- kms : CustomerMasterKeySpec (obsolète)
- kms : CustomerMasterKeyUsage (obsolète)
- km : DataKeyPairSpec
- km : EncryptionAlgorithm
- kms EncryptionContext : touche contextuelle
- km : EncryptionContextKeys
- km : ExpirationModel
- km : GrantConstraintType
- km : GrantIsFor AWSResource
- km : GrantOperations
- km : GranteePrincipal
- km : KeyOrigin
- km : KeySpec
- km : KeyUsage
- km : MacAlgorithm
- km : MessageType
- km : MultiRegion
- km : MultiRegionKeyType
- km : PrimaryRegion

- km : ReEncryptOnSameKey
- km : RequestAlias
- km : ResourceAliases
- km : ReplicaRegion
- km : RetiringPrincipal
- km : ScheduleKeyDeletionPendingWindowInDays
- km : SigningAlgorithm
- km : ValidTo
- km : ViaService
- km : WrappingAlgorithm
- km : WrappingKeySpec

km : BypassPolicyLockoutSafetyCheck

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:BypassPolicyLockoutSafetyCheck`	Booléen	À valeur unique	`CreateKey` `PutKeyPolicy`	Politiques IAM uniquement Politiques de clé et politiques IAM

La clé de `kms:BypassPolicyLockoutSafetyCheck` condition contrôle l'accès aux PutKeyPolicyopérations CreateKeyet en fonction de la valeur du `BypassPolicyLockoutSafetyCheck` paramètre dans la demande.

L'exemple d'instruction de politique IAM suivant empêche les utilisateurs de contourner le contrôle de sécurité du verrouillage de la politique en leur refusant l'autorisation de créer des clés KMS lorsque la valeur du paramètre `BypassPolicyLockoutSafetyCheck` dans la demande `CreateKey` est `true`..

```
{
  "Effect": "Deny",
```

```
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Vous pouvez également utiliser la clé de condition `kms:BypassPolicyLockoutSafetyCheck` dans une politique IAM ou une politique de clé pour contrôler l'accès à l'opération `PutKeyPolicy`. L'exemple d'instruction de politique suivant dans une politique de clé empêche les utilisateurs de contourner le contrôle de sécurité du verrouillage de la politique lors de la modification de la politique d'une clé KMS.

Plutôt que d'utiliser explicitement `Deny`, cette instruction de politique utilise `Allow` avec l'opérateur de condition Null afin d'autoriser l'accès uniquement lorsque la demande n'inclut pas le paramètre `BypassPolicyLockoutSafetyCheck`. Lorsque le paramètre n'est pas utilisé, la valeur par défaut est `false`. Cette instruction de politique légèrement affaiblie peut être remplacée dans le cas rare où un contournement est nécessaire.

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Voir aussi

- km : KeySpec
- km : KeyOrigin
- km : KeyUsage

km : CallerAccount

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:CallerAccount`	Chaîne	À valeur unique	Opérations liées aux ressources de clé KMS Opérations liées au magasin de clés personnalisé	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour autoriser ou refuser l'accès à toutes les identités (utilisateurs et rôles) d'un Compte AWS. Dans les politiques de clé, vous utilisez l'élément `Principal` pour spécifier les identités auxquelles l'instruction de politique s'applique. La syntaxe de l'élément `Principal` ne permet pas de spécifier toutes les identités dans un Compte AWS. Mais vous pouvez obtenir cet effet en combinant cette clé de condition avec un `Principal` élément qui spécifie toutes les AWS identités.

Vous pouvez l'utiliser pour contrôler l'accès à n'importe quelle opération de ressource de clé KMS, c'est-à-dire toute AWS KMS opération utilisant une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS key` dans la colonne `Resources` de l'opération. Elle est également valable pour les opérations qui gèrent des magasins de clés personnalisés.

Par exemple, l'instruction de politique suivante montre comment utiliser la clé de condition `kms:CallerAccount`. Cette déclaration de politique fait partie de la politique clé Clé gérée par AWS d'Amazon EBS. Il combine un `Principal` élément qui spécifie toutes les AWS identités avec la clé de `kms:CallerAccount` condition pour autoriser efficacement l'accès à toutes les identités dans Compte AWS 111122223333. Il contient une clé de AWS KMS condition supplémentaire (`kms:ViaService`) pour limiter davantage les autorisations en n'autorisant que les demandes provenant d'Amazon EBS. Pour de plus amples informations, veuillez consulter km : ViaService.

```
{
  "Sid": "Allow access through EBS for all principals in the account that are
 authorized to use EBS",
```

```
  "Effect": "Allow",
  "Principal": {"AWS": "*"},
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "111122223333",
      "kms:ViaService": "ec2.us-west-2.amazonaws.com"
    }
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

kms : CustomerMasterKeySpec (obsolète)

La clé de condition `kms:CustomerMasterKeySpec` est obsolète. Utilisez plutôt la clé de KeySpec condition kms :.

Les clés de condition `kms:CustomerMasterKeySpec` et `kms:KeySpec` fonctionnent de la même manière. Seuls les noms diffèrent. Nous vous recommandons d'utiliser `kms:KeySpec`. Toutefois, pour éviter d'interrompre les modifications, AWS KMS prend en charge les deux clés de condition.

kms : CustomerMasterKeyUsage (obsolète)

La clé de condition `kms:CustomerMasterKeyUsage` est obsolète. Utilisez plutôt la clé de KeyUsage condition kms :.

Les clés de condition `kms:CustomerMasterKeyUsage` et `kms:KeyUsage` fonctionnent de la même manière. Seuls les noms diffèrent. Nous vous recommandons d'utiliser `kms:KeyUsage`. Toutefois, pour éviter d'interrompre les modifications, AWS KMS prend en charge les deux clés de condition.

km : DataKeyPairSpec

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:DataKeyPairSpec`	Chaîne	À valeur unique	`GenerateDataKeyPair` `GenerateDataKeyPairWithoutPlaintext`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour contrôler l'accès aux GenerateDataKeyPairWithoutPlaintextopérations GenerateDataKeyPairet en fonction de la valeur du `KeyPairSpec` paramètre dans la demande. Par exemple, vous pouvez autoriser des utilisateurs à générer uniquement des types particuliers de paires de clés de données.

L'exemple suivant d'instruction de politique de clé utilise la clé de condition `kms:DataKeyPairSpec` pour autoriser les utilisateurs à utiliser la clé KMS afin de générer uniquement des paires de clés de données RSA.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

Voir aussi

- km : KeySpec
- the section called “km : EncryptionAlgorithm”
- the section called “kms EncryptionContext : touche contextuelle”
- the section called “km : EncryptionContextKeys”

km : EncryptionAlgorithm

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d’API	Type de stratégie

`kms:EncryptionAlgorithm`	Chaîne	À valeur unique	`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
				Politiques de clé et politiques IAM

Vous pouvez utiliser la clé de condition `kms:EncryptionAlgorithm` pour contrôler l'accès aux opérations de chiffrement en fonction de l'algorithme de chiffrement utilisé dans l'opération. Pour les ReEncryptopérations de chiffrement, de déchiffrement et de déchiffrement, il contrôle l'accès en

fonction de la valeur du EncryptionAlgorithmparamètre figurant dans la demande. Pour les opérations qui génèrent des clés de données et des paires de clés de données, elle contrôle l'accès en fonction de l'algorithme de chiffrement utilisé pour chiffrer la clé de données.

Cette clé de condition n'a aucun effet sur les opérations effectuées en dehors de AWS KMS, telles que le chiffrement avec la clé publique dans une paire de clés KMS asymétrique en dehors de. AWS KMS

EncryptionAlgorithm paramètre dans une demande

Pour autoriser les utilisateurs à utiliser uniquement un algorithme de chiffrement particulier avec une clé KMS, utilisez une instruction de politique avec un effet `Deny` et un opérateur de condition `StringNotEquals`. Par exemple, l'exemple suivant d'instruction de politique de clé interdit aux principaux pouvant endosser le rôle `ExampleRole` d'utiliser cette clé KMS dans les opérations de chiffrement spécifiées, sauf si l'algorithme de chiffrement de la demande est `RSAES_OAEP_SHA_256`. un algorithme de chiffrement asymétrique utilisé avec des clés KMS RSA.

Contrairement à une instruction de politique permettant à un utilisateur d'utiliser un algorithme de chiffrement particulier, une instruction de politique avec un double négatif comme celui-ci empêche les autres politiques et octrois associés à cette clé KMS d'autoriser ce rôle à utiliser d'autres algorithmes de chiffrement. Le paramètre `Deny` dans cette instruction de politique de clé prévaut sur toute politique de clé ou politique IAM ayant un effet `Allow`. Il prévaut également sur tous les octrois associés à cette clé KMS et à ses principaux.

```
{
  "Sid": "Allow only one encryption algorithm with this asymmetric KMS key",
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "RSAES_OAEP_SHA_256"
    }
  }
```

```
}
```

Algorithme de chiffrement utilisé pour l'opération

Vous pouvez également utiliser la clé de condition `kms:EncryptionAlgorithm` pour contrôler l'accès aux opérations en fonction de l'algorithme de chiffrement utilisé dans l'opération, même lorsque l'algorithme n'est pas spécifié dans la demande. Cela vous permet d'exiger ou d'interdire l'algorithme `SYMMETRIC_DEFAULT`, qui peut ne pas être spécifié dans une demande, car il s'agit de la valeur par défaut.

Cette fonction vous permet également d'utiliser la clé de condition `kms:EncryptionAlgorithm` pour contrôler l'accès aux opérations qui génèrent des clés de données et des paires de clés de données. Ces opérations utilisent uniquement des clés KMS de chiffrement symétriques et l'algorithme `SYMMETRIC_DEFAULT`.

Par exemple, cette politique IAM limite ses principaux au chiffrement symétrique. Elle interdit l'accès à toute clé KMS dans l'exemple de compte pour les opérations de chiffrement, sauf si l'algorithme de chiffrement spécifié dans la demande ou utilisé dans l'opération est SYMMETRIC_DEFAULT. Y compris GenerateDataKeyles `GenerateDataKey*` ajouts GenerateDataKeyWithoutPlaintextGenerateDataKeyPair,, et GenerateDataKeyPairWithoutPlaintextaux autorisations. La condition n'a aucun effet sur ces opérations, car elles utilisent toujours un algorithme de chiffrement symétrique.

```
{
  "Sid": "AllowOnlySymmetricAlgorithm",
  "Effect": "Deny",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "SYMMETRIC_DEFAULT"
    }
  }
}
```

Voir aussi

- the section called “km : MacAlgorithm”
- km : SigningAlgorithm

kms EncryptionContext : touche contextuelle

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d’API	Type de stratégie

`kms:EncryptionContext: context-key`	Chaîne	À valeur unique	`CreateGrant`
			`Encrypt`
			`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
				Politiques de clé et politiques IAM

Vous pouvez utiliser la clé de condition `kms:EncryptionContext:context-key` pour contrôler l'accès à une clé KMS de chiffrement symétrique en fonction du contexte de chiffrement d'une demande d'opération de chiffrement. Utilisez cette clé de condition pour évaluer à la fois la clé et la valeur dans la paire de contexte de chiffrement. Pour évaluer uniquement les clés de contexte de

chiffrement ou pour exiger un contexte de chiffrement indépendamment des clés ou des valeurs, utilisez la clé de EncryptionContextKeys condition kms :.

> **Note**
>
> Les valeurs clé de condition doivent respecter les règles de caractère pour les politiques de clé et les politiques IAM. Certains caractères valides dans un contexte de chiffrement ne sont pas valides dans les politiques. Vous ne pourrez peut-être pas utiliser cette clé de condition pour exprimer toutes les valeurs du contexte de chiffrement valides. Pour plus d'informations sur les règles de document de politique de clé, voir Format de politique de clé. Pour plus d'informations sur les règles du document de politique IAM, veuillez consulter Exigences relatives aux noms IAM dans le Guide de l'utilisateur IAM.

Vous ne pouvez pas spécifier de contexte de chiffrement dans une opération de chiffrement avec une clé KMS asymétrique ou une clé KMS HMAC. Les algorithmes asymétriques et les algorithmes MAC ne prennent pas en charge un contexte de chiffrement.

Pour utiliser la clé de condition kms : EncryptionContext : context-key, remplacez l'espace réservé à la clé *contextuelle par la clé contextuelle* de chiffrement. Remplacez l'espace réservé *context-value* par la valeur du contexte de chiffrement.

```
"kms:EncryptionContext:context-key": "context-value"
```

Par exemple, la clé de condition suivante spécifie un contexte de chiffrement dans lequel la clé est `AppName` et la valeur est `ExampleApp` (`AppName = ExampleApp`).

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

Il s'agit d'une clé de condition à valeur unique. La clé de la clé de condition spécifie une clé de contexte de chiffrement particulière (context-key). Bien que vous puissiez inclure plusieurs paires de contexte de chiffrement dans chaque demande d'API, la paire de contexte de chiffrement avec le paramètre context-key ne peut avoir qu'une seule valeur. Par exemple, la clé de condition `kms:EncryptionContext:Department` s'applique uniquement aux paires de contexte de chiffrement avec une clé `Department`, et toute paire de contexte de chiffrement donnée avec la clé `Department` ne peut avoir qu'une seule valeur.

N'utilisez pas d'opérateur d'ensemble avec la clé de condition `kms:EncryptionContext:context-key`. Si vous créez une instruction de politique avec une action `Allow`, la clé de condition `kms:EncryptionContext:context-key` et l'opérateur d'ensemble `ForAllValues`, la condition autorise les demandes sans contexte de chiffrement et les demandes avec des paires de contexte de chiffrement qui ne sont pas spécifiées dans la condition de politique.

> **Warning**
>
> N'utilisez pas d'opérateur d'ensemble `ForAnyValue` ou `ForAllValues` avec cette clé de condition à valeur unique. Ces opérateurs d'ensemble peuvent créer une condition de politique qui ne nécessite pas de valeurs que vous avez l'intention d'exiger et autorise les valeurs que vous avez l'intention d'interdire.
> Si vous créez ou mettez à jour une politique qui inclut un opérateur `ForAllValues` set avec la touche contextuelle kms EncryptionContext : :, AWS KMS renvoie le message d'erreur suivant :
> `OverlyPermissiveCondition:EncryptionContext: Using the ForAllValues set operator with a single-valued condition key matches requests without the specified encryption context or with an unspecified encryption context. To fix, remove ForAllValues.`

Pour exiger une paire de contexte de chiffrement particulière, utilisez la clé de condition `kms:EncryptionContext:context-key` avec l'opérateur `StringEquals`.

L'exemple d'instruction de politique de clé suivant autorise les principaux qui peuvent endosser le rôle à utiliser la clé KMS dans une demande `GenerateDataKey`, uniquement lorsque le contexte de chiffrement de la demande inclut la paire `AppName:ExampleApp`. D'autres paires de contexte de chiffrement sont autorisées.

Le nom de la clé n'est pas sensible à la casse. La sensibilité à la casse de la valeur est déterminée par l'opérateur de condition, tel que `StringEquals`. Pour plus de détails, veuillez consulter Sensibilité à la casse de la condition de contexte de chiffrement.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
```

```
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

Pour exiger une paire de contextes de chiffrement et interdire toutes les autres paires de contextes de chiffrement, utilisez à la fois la clé de contexte kms EncryptionContext : : et `kms:EncryptionContextKeys`dans la déclaration de politique. L'exemple d'instruction de politique suivant utilise la condition `kms:EncryptionContext:AppName` pour exiger la présence de la paire de contexte de chiffrement `AppName=ExampleApp` dans la demande. Il utilise également une clé de condition `kms:EncryptionContextKeys` avec l'opérateur d'ensemble `ForAllValues` pour autoriser uniquement la clé de contexte de chiffrement `AppName`.

L'opérateur d'ensemble `ForAllValues` limite les clés de contexte de chiffrement dans la demande à `AppName`. Si la condition `kms:EncryptionContextKeys` avec l'opérateur d'ensemble `ForAllValues` a été utilisée seule dans une instruction de politique, cet opérateur d'ensemble autoriserait les demandes sans contexte de chiffrement. Toutefois, si la demande n'avait pas de contexte de chiffrement, la condition `kms:EncryptionContext:AppName` échouerait. Pour plus de détails sur l'opérateur d'ensemble `ForAllValues`, veuillez consulter Utilisation de plusieurs clés et valeurs dans le Guide de l'utilisateur IAM.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KeyUsers"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "AppName"
      ]
    }
```

```
  }
}
```

Vous pouvez également utiliser cette clé de condition pour refuser l'accès à une clé KMS pour une opération particulière. L'exemple d'instruction de politique de clé suivant utilise un effet `Deny` pour interdire au principal d'utiliser la clé KMS si le contexte de chiffrement de la demande inclut une paire de contexte de chiffrement `Stage=Restricted`. Cette condition permet une demande avec d'autres paires de contexte de chiffrement, y compris les paires de contexte de chiffrement avec la clé `Stage` et d'autres valeurs, telles que `Stage=Test`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

Utilisation de plusieurs paires de contexte de chiffrement

Vous pouvez exiger ou interdire plusieurs paires de contexte de chiffrement. Vous pouvez également exiger l'une des différentes paires de contexte de chiffrement. Pour plus de détails sur la logique utilisée pour interpréter ces conditions, veuillez consulter Création d'une condition avec plusieurs clés ou valeurs dans le Guide de l'utilisateur IAM.

> **Note**
>
> Les versions antérieures de cette rubrique affichaient des déclarations de politique qui utilisaient les opérateurs `ForAnyValue` et `ForAllValues` set avec la clé de condition kms EncryptionContext : : context-key. L'utilisation d'un opérateur d'ensemble avec une clé de condition à valeur unique peut entraîner des politiques qui autorisent des demandes sans contexte de chiffrement et des paires de contexte de chiffrement non spécifiées.
> Par exemple, une condition de politique avec l'effet `Allow`, l'opérateur d'ensemble `ForAllValues` et la clé de condition `"kms:EncryptionContext:Department": "IT"`

ne limite pas le contexte de chiffrement à la paire « Department = IT ». Elle autorise les demandes sans contexte de chiffrement et les demandes avec des paires de contexte de chiffrement non spécifiées, telles que `Stage=Restricted`.
Veuillez revoir vos politiques et éliminer l'opérateur défini de toute condition à l'aide de la touche contextuelle kms EncryptionContext : :. Les tentatives de création ou de mise à jour d'une politique avec ce format échouent avec une exception `OverlyPermissiveCondition`. Pour résoudre l'erreur, supprimez l'opérateur d'ensemble.

Pour exiger plusieurs paires de contexte de chiffrement, répertoriez les paires dans la même condition. L'exemple d'instruction de politique de clé suivant nécessite deux paires de contexte de chiffrement, `Department=IT` et `Project=Alpha`. Puisque les conditions ont des clés différentes (`kms:EncryptionContext:Department` et `kms:EncryptionContext:Project`), elles sont implicitement connectées par un opérateur AND. D'autres paires de contexte de chiffrement sont autorisées, mais non exigées.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    }
  }
}
```

Pour exiger une paire de contexte de chiffrement OU une autre paire, placez chaque clé de condition dans une instruction de politique distincte. L'exemple de politique de clé suivant nécessite des paires `Department=IT` ou `Project=Alpha`, ou les deux. D'autres paires de contexte de chiffrement sont autorisées, mais non exigées.

```
{
 "Effect": "Allow",
 "Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
```

```
 },
 "Action": "kms:GenerateDataKey",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "kms:EncryptionContext:Department": "IT"
  }
 }
},
{
 "Effect": "Allow",
 "Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
 },
 "Action": "kms:GenerateDataKey",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "kms:EncryptionContext:Project": "Alpha"
  }
 }
}
```

Pour exiger des paires de chiffrement particulières et exclure toutes les autres paires de contextes de chiffrement, utilisez à la fois la clé de contexte kms EncryptionContext : : et kms:EncryptionContextKeysdans la déclaration de politique. La déclaration de politique clé suivante utilise la condition de clé de contexte kms EncryptionContext : : pour exiger un contexte de chiffrement avec `Department=IT` les `Project=Alpha` deux paires. Elle utilise une clé de condition `kms:EncryptionContextKeys` avec l'opérateur d'ensemble `ForAllValues` pour n'autoriser que les clés de contexte de chiffrement `Department` et `Project`.

L'opérateur d'ensemble `ForAllValues` limite les clés de contexte de chiffrement dans la demande à `Department` et `Project`. S'il était utilisé seul dans une condition, cet opérateur set autoriserait les requêtes sans contexte de chiffrement, mais dans cette configuration, la clé de contexte kms EncryptionContext : : dans cette condition échouerait.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
```

```
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "Department",
        "Project"
      ]
    }
  }
}
```

Vous pouvez également interdire plusieurs paires de contexte de chiffrement. L'exemple d'instruction de politique de clé suivant utilise un effet `Deny` pour interdire au principal d'utiliser les clés KMS si le contexte de chiffrement de la demande inclut une paire `Stage=Restricted` ou `Stage=Production`.

Plusieurs valeurs (`Restricted` et `Production`) pour la même clé (`kms:EncryptionContext:Stage`) sont implicitement connectés par un OR. Pour plus de détails, veuillez consulter Logique d'évaluation pour les conditions avec plusieurs clés ou valeurs dans le Guide de l'utilisateur IAM.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
         "Restricted",
         "Production"
      ]
    }
  }
}
```

Sensibilité à la casse de la condition de contexte de chiffrement

Le contexte de chiffrement indiqué dans une opération de déchiffrement doit correspondre exactement au contexte de chiffrement précisé dans l'opération de chiffrement (en tenant compte des minuscules/majuscules). Seul l'ordre des paires dans un contexte de chiffrement avec plusieurs paire peut varier.

En revanche, dans les conditions des politiques, la clé de condition n'est pas sensible à la casse. Le respect de la casse de la valeur de condition est déterminée par l'opérateur de condition de politique que vous utilisez, comme `StringEquals` ou `StringEqualsIgnoreCase`.

À ce titre, la clé de condition, qui comprend le préfixe `kms:EncryptionContext:` et le remplacement *`context-key`*, n'est pas sensible à la casse. Une politique qui utilise cette condition ne vérifie pas la casse des éléments de la clé de condition. Le respect de la casse de la valeur, à savoir, le remplacement *`context-value`*, est déterminé par l'opérateur de condition de la politique.

Par exemple, l'instruction de politique suivante autorise l'opération lorsque le contexte de chiffrement inclut une clé `Appname`, quelle que soit sa capitalisation. La condition `StringEquals` nécessite que `ExampleApp` soit capitalisé tel qu'il est spécifié.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Appname": "ExampleApp"
    }
  }
}
```

Pour exiger une clé contextuelle de chiffrement distinguant majuscules et minuscules, utilisez la condition kms : EncryptionContextKeys policy avec un opérateur de condition sensible aux majuscules et minuscules, tel que. `StringEquals` Dans cette condition de politique, étant donné que la clé de contexte de chiffrement est la valeur de cette condition de politique, sa sensibilité à la casse est déterminée par l'opérateur de condition.

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Pour exiger une évaluation distinguant majuscules et minuscules de la clé et de la valeur du contexte de chiffrement, utilisez les conditions de politique de clé de contexte *`kms:EncryptionContextKeys`* et kms EncryptionContext : : ensemble dans la même déclaration de politique. L'opérateur de condition sensible à la casse (tel que `StringEquals`) s'applique toujours à la valeur de la condition. La clé de contexte de chiffrement (telle que AppName) est la valeur de la condition `kms:EncryptionContextKeys`. La valeur du contexte de chiffrement (telle que`ExampleApp`) est la valeur de la condition de clé de contexte kms EncryptionContext : :.

Par exemple, dans l'exemple suivant d'instruction de politique de clé, étant donné que l'opérateur `StringEquals` est sensible à la casse, la clé de contexte de chiffrement et la valeur de contexte de chiffrement sont toutes deux sensibles à la casse.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    },
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

Utilisation de variables dans une condition de contexte de chiffrement

La clé et la valeur d'une paire de contexte de chiffrement doivent être des chaînes littérales simples. Il ne peut pas s'agir d'entiers ou d'objets, ou d'un type qui n'est pas entièrement résolu. Si vous utilisez un autre type, tel qu'un entier ou un flottant, il est AWS KMS interprété comme une chaîne littérale.

```
"encryptionContext": {
    "department": "10103.0"
}
```

Toutefois, la valeur de la clé de condition `kms:EncryptionContext:`*`context-key`* peut être une variable de politique IAM. Ces variables de politique sont résolues lors de l'exécution en fonction des valeurs de la demande. Par exemple, `aws:CurrentTime` est résolue à l'heure de la demande et `aws:username` est résolue au nom convivial de l'appelant.

Vous pouvez utiliser ces variables de politique pour créer une instruction de politique avec une condition qui nécessite des informations très spécifiques dans un contexte de chiffrement, comme le nom d'utilisateur de l'appelant. Comme elle contient une variable, vous pouvez utiliser la même instruction de politique pour tous les utilisateurs qui peuvent assumer le rôle. Vous n'avez pas besoin d'écrire une instruction de politique distincte pour chaque utilisateur.

Prenons un exemple : vous voulez que tous les utilisateurs qui peuvent endosser un rôle utilisent la même clé KMS pour chiffrer et déchiffrer leurs données. Cependant, vous souhaitez leur permettre de déchiffrer uniquement les données qu'ils ont chiffrées. Commencez par exiger que chaque demande AWS KMS inclue un contexte de chiffrement dans lequel la clé est le nom d'utilisateur de l'appelant `user` et dont la valeur est le nom AWS d'utilisateur, tel que le suivant.

```
"encryptionContext": {
    "user": "bob"
}
```

Ensuite, pour appliquer cette exigence, vous pouvez utiliser une instruction de politique comme celle de l'exemple suivant. Cette instruction de politique accorde au rôle `TestTeam` l'autorisation de chiffrer et de déchiffrer les données avec la clé KMS. Toutefois, l'autorisation est uniquement valable lorsque le contexte de chiffrement de la demande inclut une paire `"user": "`*`<username>`*`"`. Pour représenter le nom d'utilisateur, la condition utilise la variable de politique `aws:username`.

Lorsque la demande est évaluée, le nom d'utilisateur de l'appelant remplace la variable dans la condition. Ainsi, la condition nécessite un contexte de chiffrement `"user": "bob"` pour « bob » et `"user": "alice"` pour « alice ».

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:user": "${aws:username}"
    }
  }
}
```

Vous pouvez utiliser une variable de politique IAM uniquement dans la valeur de la clé de condition `kms:EncryptionContext:`*`context-key`*. Vous ne pouvez pas utiliser une variable dans la clé.

Vous pouvez également utiliser des clés de contexte spécifiques au fournisseur dans des variables. Ces clés contextuelles identifient de manière unique les utilisateurs qui se sont connectés à AWS l'aide de la fédération d'identité Web.

Comme toutes les variables, celles-ci peuvent être utilisées uniquement dans la condition de politique `kms:EncryptionContext:`*`context-key`*, et non dans le contexte de chiffrement réel. Et elles peuvent être utilisées uniquement dans la valeur de la condition, pas dans la clé.

Par exemple, l'instruction de politique de clé suivante est similaire à la précédente. Toutefois, la condition nécessite un contexte de chiffrement où la clé est `sub` et la valeur identifie de façon unique un utilisateur connecté à un groupe d'utilisateurs Amazon Cognito. Pour plus de détails sur l'identification des utilisateurs et les rôles dans Amazon Cognito, veuillez consulter Rôles IAM dans le guide du développeur Amazon Cognito.

```
{
  "Effect": "Allow",
  "Principal": {
```

```
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
       "kms:EncryptionContext:sub": "${cognito-identity.amazonaws.com:sub}"
    }
  }
}
```

Voir aussi

- the section called “km : EncryptionContextKeys”
- the section called “km : GrantConstraintType”

km : EncryptionContextKeys

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d’API	Type de stratégie

`kms:EncryptionContextKeys`	Chaîne (liste)	Multi-valeurs	`CreateGrant`
			`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
				Politiques de clé et politiques IAM

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie

			`GenerateDataKeyWithoutPlaintext`
			
			`ReEncrypt`	

Vous pouvez utiliser la clé de condition `kms:EncryptionContextKeys` pour contrôler l'accès à une clé KMS de chiffrement symétrique en fonction du contexte de chiffrement d'une demande d'opération de chiffrement. Utilisez cette clé de condition pour évaluer uniquement la clé dans chaque paire de contexte de chiffrement. Utilisez la clé de condition `kms:EncryptionContext:`*`context-key`* afin d'évaluer à la fois la clé et la valeur dans le contexte de chiffrement.

Vous ne pouvez pas spécifier de contexte de chiffrement dans une opération de chiffrement avec une clé KMS asymétrique ou une clé KMS HMAC. Les algorithmes asymétriques et les algorithmes MAC ne prennent pas en charge un contexte de chiffrement.

> **Note**
>
> Les valeurs des clés de condition, y compris une clé de contexte de chiffrement, doivent être conformes aux règles de caractères et de codage des politiques AWS KMS clés. Vous ne pourrez peut-être pas utiliser cette clé de condition pour exprimer toutes les clés du contexte de chiffrement valides. Pour plus d'informations sur les règles de document de politique de clé, voir Format de politique de clé. Pour plus d'informations sur les règles du document de politique IAM, veuillez consulter Exigences relatives aux noms IAM dans le Guide de l'utilisateur IAM.

Il s'agit d'une clé de condition multi-valeurs. Vous pouvez spécifier plusieurs paires de contexte de chiffrement dans chaque demande d'API. `kms:EncryptionContextKeys` compare les clés de contexte de chiffrement dans la demande à l'ensemble des clés de contexte de chiffrement dans la politique. Pour déterminer comment ces ensembles sont comparés, vous devez fournir un opérateur d'ensemble `ForAnyValue` ou `ForAllValues` dans la condition de politique. Pour plus de détails sur

les opérateurs d'ensemble, veuillez consulter Utilisation de plusieurs clés et valeurs dans le Guide de l'utilisateur IAM.

- `ForAnyValue` : au moins une clé de contexte de chiffrement dans la demande doit correspondre à une clé de contexte de chiffrement dans la condition de politique. D'autres clés de contexte de chiffrement sont autorisées. Si la demande n'a aucun contexte de chiffrement, la condition n'est pas remplie.
- `ForAllValues` : chaque clé de contexte de chiffrement de la demande doit correspondre à une clé de contexte de chiffrement dans la condition de politique. Cet opérateur d'ensemble limite les clés de contexte de chiffrement à celles de la condition de politique. Il ne nécessite aucune clé de contexte de chiffrement, mais il interdit les clés de contexte de chiffrement non spécifiées.

L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:EncryptionContextKeys` avec l'opérateur d'ensemble `ForAnyValue`. Cette instruction de politique autorise l'utilisation d'une clé KMS pour les opérations spécifiées, mais uniquement si au moins une des paires de contexte de chiffrement de la demande inclut la clé `AppName`, peu importe sa valeur.

Par exemple, cette instruction de politique de clé autorise une demande `GenerateDataKey` avec deux paires de contexte de chiffrement, `AppName=Helper` et `Project=Alpha`, car la première paire de contexte de chiffrement répond à la condition. Une demande avec uniquement `Project=Alpha` ou sans contexte de chiffrement échouerait.

Comme l'opération StringEqualsconditionnelle distingue les majuscules et minuscules, cette déclaration de politique requiert l'orthographe et les majuscules de la clé contextuelle de chiffrement. Toutefois, vous pouvez utiliser un opérateur de condition qui ignore la casse de la clé, par exemple `StringEqualsIgnoreCase`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
```

```
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Vous pouvez également utiliser la clé de condition `kms:EncryptionContextKeys` pour exiger un contexte de chiffrement (tout contexte de chiffrement) dans les opérations de cryptographiques qui utilisent la clé KMS.

L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:EncryptionContextKeys` avec l'opérateur de condition Null pour autoriser l'accès à une clé KMS uniquement lorsque le contexte de chiffrement de la demande d'API n'est pas nul. Cette condition ne vérifie pas les clés ou les valeurs du contexte de chiffrement. Elle vérifie uniquement que le contexte de chiffrement existe.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContextKeys": false
    }
  }
}
```

Voir aussi

- kms EncryptionContext : touche contextuelle
- km : GrantConstraintType

km : ExpirationModel

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:ExpirationModel`	Chaîne	À valeur unique	`ImportKeyMaterial`	Politiques de clé et politiques IAM

La clé de `kms:ExpirationModel` condition contrôle l'accès à l'ImportKeyMaterialopération en fonction de la valeur du ExpirationModelparamètre dans la demande.

`ExpirationModel` est un paramètre facultatif qui détermine si les éléments de clé importés arrivent à expiration. Les valeurs valides sont `KEY_MATERIAL_EXPIRES` et `KEY_MATERIAL_DOES_NOT_EXPIRE`. La valeur par défaut est `KEY_MATERIAL_EXPIRES`.

La date et l'heure d'expiration sont déterminées par la valeur du ValidToparamètre. Le paramètre `ValidTo` est obligatoire, sauf si la valeur du paramètre `ExpirationModel` est `KEY_MATERIAL_DOES_NOT_EXPIRE`. Vous pouvez également utiliser la clé de ValidTo condition kms : pour exiger une date d'expiration particulière comme condition d'accès.

L'exemple d'instruction de politique suivant utilise la clé de condition `kms:ExpirationModel` pour autoriser les utilisateurs à importer les éléments de clé dans une clé KMS uniquement lorsque la demande inclut le paramètre `ExpirationModel` et que sa valeur est `KEY_MATERIAL_DOES_NOT_EXPIRE`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE"
    }
  }
}
```

Vous pouvez également utiliser la clé de condition `kms:ExpirationModel` pour autoriser les utilisateurs à importer les éléments de clé uniquement lorsque ceux-ci expirent. L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:ExpirationModel` avec l'opérateur de condition Null pour autoriser les utilisateurs à importer les éléments de clé uniquement lorsque la demande ne dispose pas de paramètre `ExpirationModel`. La valeur par défaut pour ExpirationModel est`KEY_MATERIAL_EXPIRES`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

Voir aussi

- km : ValidTo
- km : WrappingAlgorithm
- km : WrappingKeySpec

km : GrantConstraintType

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:GrantConstraintType`	Chaîne	À valeur unique	`CreateGrant`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'CreateGrantopération en fonction du type de contrainte d'autorisation figurant dans la demande.

Lorsque vous créez un octroi, vous pouvez éventuellement spécifier une contrainte d'octroi pour autoriser les opérations permises par l'octroi seulement lorsqu'un contexte de chiffrement particulier est présent. La contrainte d'octroi peut être de deux types : `EncryptionContextEquals` ou `EncryptionContextSubset`. Vous pouvez utiliser cette clé de condition pour vérifier que la demande contient un type ou l'autre.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:GrantConstraintType` pour autoriser les utilisateurs à créer des octrois uniquement lorsque la demande inclut une contrainte d'octroi `EncryptionContextEquals`. L'exemple suivant montre une instruction de politique dans une politique de clé.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}
```

Voir aussi

- kms EncryptionContext : touche contextuelle
- km : EncryptionContextKeys
- km : GrantIsFor AWSResource
- km : GrantOperations
- km : GranteePrincipal
- km : RetiringPrincipal

km : GrantIsFor AWSResource

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie

`kms:GrantIsForAWSResource`	Booléen	À valeur unique	`CreateGrant`
			`ListGrants`
			`RevokeGrant`
				Politiques de clé et politiques IAM

Autorise ou refuse l'autorisation pour les RevokeGrantopérations CreateGrantListGrants, ou uniquement lorsqu'un AWS service intégré AWS KMS appelle l'opération au nom de l'utilisateur. Cette condition de politique ne permet pas à l'utilisateur d'appeler ces opérations d'octroi directement.

L'exemple suivant d'instruction de politique de clé utilise la clé de condition `kms:GrantIsForAWSResource`. Il permet aux AWS services intégrés AWS KMS, tels qu'Amazon EBS, de créer des subventions sur cette clé KMS au nom du principal spécifié.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

Voir aussi

- km : GrantConstraintType
- km : GrantOperations
- km : GranteePrincipal
- km : RetiringPrincipal

km : GrantOperations

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:GrantOperations`	Chaîne	Multi-valeurs	`CreateGrant`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'CreateGrantopération en fonction des opérations d'autorisation figurant dans la demande. Par exemple, vous pouvez autoriser les utilisateurs à créer des octrois qui délèguent l'autorisation de chiffrer mais pas de déchiffrer. Pour plus d'informations sur les octrois, veuillez consulter Utilisation d'octrois.

Il s'agit d'une clé de condition multi-valeurs. `kms:GrantOperations` compare l'ensemble d'opérations d'octrois dans la demande `CreateGrant` à l'ensemble d'opérations d'octrois dans la politique. Pour déterminer comment ces ensembles sont comparés, vous devez fournir un opérateur d'ensemble `ForAnyValue` ou `ForAllValues` dans la condition de politique. Pour plus de détails sur les opérateurs d'ensemble, veuillez consulter Utilisation de plusieurs clés et valeurs dans le Guide de l'utilisateur IAM.

- `ForAnyValue` : au moins une opération d'octroi dans la demande doit correspondre à l'une des opérations d'octrois dans la condition de politique. D'autres opérations d'octrois sont autorisées.
- ForAllValues: Chaque opération de subvention figurant dans la demande doit correspondre à une opération de subvention figurant dans la condition de politique. Cet opérateur d'ensemble limite les opérations d'octrois à celles spécifiées dans la condition de politique. Il ne nécessite aucune opération d'octroi, mais il interdit les opérations d'octrois non spécifiées.

  ForAllValues renvoie également true lorsqu'il n'y a aucune opération de subvention dans la demande, mais `CreateGrant` ne l'autorise pas. Si le paramètre `Operations` manque ou qu'il a une valeur nulle, la demande `CreateGrant` échoue.

L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:GrantOperations` pour créer des octrois uniquement lorsque les opérations d'octrois sont `Encrypt`, `ReEncryptTo` ou les deux. Si l'octroi inclut toute autre opération, la demande `CreateGrant` échoue.

```
{
  "Effect": "Allow",
```

```
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```

Si vous changez l'opérateur d'ensemble dans la condition de politique en `ForAnyValue`, l'instruction de politique exigerait qu'au moins une des opérations d'octrois dans l'octroi soit `Encrypt` ou `ReEncryptTo`, mais elle autoriserait d'autres opérations d'octrois, telles que `Decrypt` ou `ReEncryptFrom`.

Voir aussi

- km : GrantConstraintType
- km : GrantIsFor AWSResource
- km : GranteePrincipal
- km : RetiringPrincipal

km : GranteePrincipal

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:GranteePrincipal`	Chaîne	À valeur unique	`CreateGrant`	Politiques de clé et IAM

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'CreateGrantopération en fonction de la valeur du GranteePrincipalparamètre dans la demande. Par exemple, vous pouvez créer

des octrois pour utiliser une clé KMS uniquement lorsque le principal bénéficiaire de la demande `CreateGrant` correspond au principal spécifié dans l'instruction de condition.

Pour spécifier le principal du bénéficiaire, utilisez le Amazon Resource Name (ARN) d'un AWS principal. Les principaux valides incluent les utilisateurs IAM Comptes AWS, les rôles IAM, les utilisateurs fédérés et les utilisateurs des rôles assumés. Pour obtenir de l'aide sur la syntaxe ARN d'un principal, consultez la section ARN IAM dans le guide de l'utilisateur IAM.

L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:GranteePrincipal` afin de créer des octrois pour une clé KMS uniquement lorsque le principal bénéficiaire de l'octroi est le `LimitedAdminRole`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GranteePrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

Voir aussi

- km : GrantConstraintType
- km : GrantIsFor AWSResource
- km : GrantOperations
- km : RetiringPrincipal

km : KeyOrigin

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:KeyOrigin`	Chaîne	À valeur unique	`CreateKey` Opérations liées aux ressources de clé KMS	Politiques IAM Politiques de clé et politiques IAM

La clé de condition `kms:KeyOrigin` contrôle l'accès aux opérations en fonction de la valeur de la propriété `Origin` de la clé KMS créée par l'opération ou utilisée dans cette dernière. Elle fonctionne comme une condition de ressource ou une condition de demande.

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'CreateKeyopération en fonction de la valeur du paramètre Origin dans la demande. Les valeurs valides pour `Origin` sont `AWS_KMS`, `AWS_CLOUDHSM` et `EXTERNAL`.

Par exemple, vous pouvez créer une clé KMS uniquement lorsque le contenu clé est généré dans AWS KMS (`AWS_KMS`), uniquement lorsque le matériau clé est généré dans un AWS CloudHSM cluster associé à un magasin de clés personnalisé (`AWS_CLOUDHSM`), ou uniquement lorsque le matériau clé est importé depuis une source externe (`EXTERNAL`).

L'exemple de déclaration de politique clé suivant utilise la clé de `kms:KeyOrigin` condition pour créer une clé KMS uniquement lors de la AWS KMS création du matériel clé.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": "kms:CreateKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```
          "kms:KeyOrigin": "AWS_KMS"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:GenerateDataKeyPair",
        "kms:GenerateDataKeyPairWithoutPlaintext",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_CLOUDHSM"
        }
      }
    }
  ]
}
```

Vous pouvez également utiliser la clé de condition `kms:KeyOrigin` pour contrôler l'accès aux opérations qui utilisent ou gèrent une clé KMS en fonction de la propriété `Origin` de la clé KMS utilisée pour l'opération. L'opération doit être une opération de ressource de clé KMS, c'est-à-dire une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS key` dans la colonne `Resources` de l'opération.

Par exemple, la politique IAM suivante permet aux principaux d'effectuer les opérations de ressources de clé KMS spécifiées, mais uniquement avec les clés KMS du compte qui ont été créées dans un magasin de clés personnalisé.

```
{
  "Effect": "Allow",
  "Action": [
```

```
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext",
    "kms:ReEncrypt*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_CLOUDHSM"
    }
  }
}
```

Voir aussi

- km : BypassPolicyLockoutSafetyCheck
- km : KeySpec
- km : KeyUsage

km : KeySpec

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:KeySpec`	Chaîne	À valeur unique	`CreateKey` Opérations liées aux ressources de clé KMS	Politiques IAM Politiques de clé et politiques IAM

La clé de condition `kms:KeySpec` contrôle l'accès aux opérations en fonction de la valeur de la propriété `KeySpec` de la clé KMS créée par l'opération ou utilisée dans cette dernière.

Vous pouvez utiliser cette clé de condition dans une politique IAM pour contrôler l'accès à l'CreateKeyopération en fonction de la valeur du KeySpecparamètre dans une `CreateKey` demande.

Par exemple, vous pouvez utiliser cette condition pour autoriser les utilisateurs à créer uniquement des clés KMS de chiffrement symétriques ou des clés KMS HMAC.

L'exemple d'instruction de politique IAM suivant utilise la clé de condition `kms:KeySpec` pour autoriser les principaux à créer des clés KMS asymétriques RSA uniquement. L'autorisation n'est valide que lorsque la valeur `KeySpec` dans la demande commence par `RSA_`.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:KeySpec": "RSA_*"
    }
  }
}
```

Vous pouvez également utiliser la clé de condition `kms:KeySpec` pour contrôler l'accès aux opérations qui utilisent ou gèrent une clé KMS en fonction de la propriété `KeySpec` de la clé KMS utilisée pour l'opération. L'opération doit être une opération de ressource de clé KMS, c'est-à-dire une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS key` dans la colonne `Resources` de l'opération.

Par exemple, la politique IAM suivante permet aux principaux d'effectuer les opérations de ressources de clé KMS spécifiées, mais uniquement avec les clés KMS de chiffrement symétriques du compte.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeySpec": "SYMMETRIC_DEFAULT"
```

```
    }
  }
}
```

Voir aussi

- km : BypassPolicyLockoutSafetyCheck
- kms : CustomerMasterKeySpec (obsolète)
- km : DataKeyPairSpec
- km : KeyOrigin
- km : KeyUsage

km : KeyUsage

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:KeyUsage`	Chaîne	À valeur unique	`CreateKey` Opérations liées aux ressources de clé KMS	Politiques IAM Politiques de clé et politiques IAM

La clé de condition `kms:KeyUsage` contrôle l'accès aux opérations en fonction de la valeur de la propriété `KeyUsage` de la clé KMS créée par l'opération ou utilisée dans cette dernière.

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'CreateKeyopération en fonction de la valeur du KeyUsageparamètre dans la demande. Les valeurs valides pour `KeyUsage` sont `ENCRYPT_DECRYPT`, `SIGN_VERIFY` et `GENERATE_VERIFY_MAC`.

Par exemple, vous pouvez créer une clé KMS uniquement lorsque `KeyUsage` a pour valeur `ENCRYPT_DECRYPT` ou refuser à un utilisateur cette autorisation lorsque `KeyUsage` a pour valeur `SIGN_VERIFY`.

L'exemple d'instruction de politique IAM suivant utilise la clé de condition `kms:KeyUsage` pour créer une clé KMS uniquement lorsque le paramètre `KeyUsage` a pour valeur `ENCRYPT_DECRYPT`.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "ENCRYPT_DECRYPT"
    }
  }
}
```

Vous pouvez également utiliser la clé de condition `kms:KeyUsage` pour contrôler l'accès aux opérations qui utilisent ou gèrent une clé KMS en fonction de la propriété `KeyUsage` de la clé KMS utilisée pour l'opération. L'opération doit être une opération de ressource de clé KMS, c'est-à-dire une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS key` dans la colonne `Resources` de l'opération.

Par exemple, la politique IAM suivante permet aux principaux d'effectuer les opérations de ressources de clé KMS spécifiées, mais uniquement avec les clés KMS du compte qui sont utilisées pour la signature et la vérification.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant",
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:ScheduleKeyDeletion"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "SIGN_VERIFY"
    }
  }
}
```

Voir aussi

- km : BypassPolicyLockoutSafetyCheck

- kms : CustomerMasterKeyUsage (obsolète)
- km : KeyOrigin
- km : KeySpec

km : MacAlgorithm

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:MacAlgorithm`	Chaîne	À valeur unique	`GenerateMac` `VerifyMac`	Politiques de clé et politiques IAM

Vous pouvez utiliser la clé de `kms:MacAlgorithm` condition pour contrôler l'accès aux VerifyMacopérations GenerateMacet en fonction de la valeur du `MacAlgorithm` paramètre dans la demande.

L'exemple de politique de clé suivant permet aux utilisateurs qui peuvent endosser le rôle `testers` d'utiliser la clé KMS HMAC pour générer et vérifier les balises HMAC uniquement lorsque l'algorithme MAC de la requête est `HMAC_SHA_384` ou `HMAC_SHA_512`. Cette politique utilise deux instructions de politique distinctes ayant chacune leur propre condition. Si vous spécifiez plusieurs algorithmes MAC dans une seule instruction de condition, la condition nécessite les deux algorithmes, au lieu de l'un ou de l'autre.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_384"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_512"
        }
      }
    }
  ]
}
```

Voir aussi

- the section called “km : EncryptionAlgorithm”
- km : SigningAlgorithm

km : MessageType

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d’API	Type de stratégie
`kms:MessageType`	Chaîne	À valeur unique	`Sign` `Verify`	Politiques de clé et politiques IAM

La clé de condition `kms:MessageType` contrôle l'accès aux opérations Sign et Verify en fonction de la valeur du paramètre `MessageType` de la demande. Les valeurs valides pour `MessageType` sont `RAW` et `DIGEST`.

Par exemple, l'instruction de politique de clé suivante utilise la clé de condition `kms:MessageType` afin d'utiliser une clé KMS asymétrique pour signer un message, mais pas un condensé du message.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:MessageType": "RAW"
    }
  }
}
```

Voir aussi

- the section called “km : SigningAlgorithm”

km : MultiRegion

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d’API	Type de stratégie
`kms:MultiRegion`	Booléen	À valeur unique	`CreateKey` Opérations liées aux ressources de clé KMS	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour autoriser les opérations uniquement sur des clés à région unique ou uniquement sur des clés multi-région. La clé de `kms:MultiRegion` condition contrôle l'accès aux AWS KMS opérations sur les clés KMS et à l'CreateKeyopération en fonction de

la valeur de la `MultiRegion` propriété de la clé KMS. Les valeurs valides sont `true` (multi-région) et `false` (région unique). Toutes les clés KMS ont une propriété `MultiRegion`.

Par exemple, l'instruction de politique IAM suivante utilise la clé de condition `kms:MultiRegion` afin d'autoriser les principaux à créer uniquement des clés à région unique.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}
```

km : MultiRegionKeyType

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:MultiRegionKeyType`	Chaîne	À valeur unique	`CreateKey` Opérations liées aux ressources de clé KMS	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour autoriser les opérations uniquement sur des clés primaires multi-région ou uniquement sur des clés de réplica multi-région. La clé de `kms:MultiRegionKeyType` condition contrôle l'accès aux AWS KMS opérations sur les clés KMS et l'CreateKeyopération en fonction de la `MultiRegionKeyType` propriété de la clé KMS. Les valeurs valides sont `PRIMARY` et `REPLICA`. Seules les clés multi-région ont une propriété `MultiRegionKeyType`.

En général, vous utilisez la clé de condition `kms:MultiRegionKeyType` dans une politique IAM pour contrôler l'accès à plusieurs clés KMS. Toutefois, comme une clé multi-région donnée peut se changer en primaire ou en réplica, vous devrez peut-être utiliser cette condition dans une politique

de clé pour autoriser une opération uniquement lorsque la clé multi-région particulière est une clé primaire ou réplica.

Par exemple, l'instruction de politique IAM suivante utilise la clé de condition `kms:MultiRegionKeyType` pour permettre aux principaux de planifier et d'annuler la suppression des clés uniquement sur les clés de réplica multi-région dans le Compte AWSspécifié.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:MultiRegionKeyType": "REPLICA"
    }
  }
}
```

Pour autoriser ou refuser l'accès à toutes les clés multi-région, vous pouvez utiliser les deux valeurs ou une valeur nulle avec `kms:MultiRegionKeyType`. Cependant, la clé de MultiRegion condition kms : est recommandée à cette fin.

km : PrimaryRegion

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:PrimaryRegion`	Chaîne (liste)	À valeur unique	`UpdatePrimaryRegion`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour limiter les régions de destination dans une UpdatePrimaryRegionopération. Ils Régions AWS peuvent héberger vos clés primaires multirégionales.

La touche de `kms:PrimaryRegion` condition contrôle l'accès à l'UpdatePrimaryRegionopération en fonction de la valeur du `PrimaryRegion` paramètre. Le `PrimaryRegion` paramètre spécifie la

clé Région AWS de réplique multirégionale qui est promue au rang de clé principale. La valeur de la condition est un ou plusieurs Région AWS noms, tels que `us-east-1` ou`ap-southeast-2`, ou des modèles de noms de région, tels que `eu-*`

Par exemple, l'instruction de politique de clé suivante utilise la clé de condition `kms:PrimaryRegion` afin de permettre aux principaux de mettre à jour la région primaire d'une clé multi-région vers une des quatre régions spécifiées.

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Developer"
  },
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-east-1",
         "us-west-2",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

km : ReEncryptOnSameKey

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:ReEncryptOnSameKey`	Booléen	À valeur unique	`ReEncrypt`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'ReEncryptopération selon que la demande spécifie ou non une clé KMS de destination identique à celle utilisée pour le chiffrement d'origine.

Par exemple, l'instruction de politique de clé suivante utilise la clé de condition `kms:ReEncryptOnSameKey` pour rechiffrer uniquement lorsque la clé KMS de destination est identique à celle utilisée pour le chiffrement d'origine.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:ReEncryptOnSameKey": true
    }
  }
}
```

km : RequestAlias

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:RequestAlias`	Chaîne (liste)	À valeur unique	Opérations cryptographiques DescribeKey GetPublicKey	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour autoriser une opération uniquement lorsque la demande utilise un alias particulier pour identifier la clé KMS. La clé de condition `kms:RequestAlias` contrôle l'accès à une clé KMS utilisée lors d'une opération cryptographique, `GetPublicKey`, ou `DescribeKey` en fonction de l'alias qui identifie cette clé KMS dans la demande. (Cette condition de politique n'a aucun effet sur l'GenerateRandomopération car celle-ci n'utilise pas de clé ou d'alias KMS.)

Cette condition prend en charge le contrôle d'accès basé sur les attributs (ABAC) dans AWS KMS, qui vous permet de contrôler l'accès aux clés KMS en fonction des balises et des alias d'une clé

KMS. Vous pouvez utiliser des balises et des alias pour autoriser ou refuser l'accès à une clé KMS sans modifier les politiques ou les octrois. Pour plus de détails, veuillez consulter ABAC pour AWS KMS.

Pour spécifier l'alias dans cette condition de politique, utilisez un nom d'alias, tel que `alias/project-alpha`, ou un modèle de nom d'alias, tel que `alias/*test*`. Vous ne pouvez pas spécifier un ARN d'alias dans la valeur de cette clé de condition.

Pour satisfaire à cette condition, la valeur du paramètre `KeyId` dans la demande doit être un nom d'alias ou un ARN d'alias correspondant. Si la demande utilise un identifiant de clé, elle ne satisfait pas à la condition, même si elle identifie la même clé KMS.

Par exemple, la déclaration de politique clé suivante permet au principal d'appeler l'GenerateDataKeyopération sur la clé KMS. Cependant, cela n'est autorisé que lorsque la valeur du paramètre `KeyId` dans la demande est `alias/finance-key` ou un ARN d'alias avec ce nom d'alias, tel que `arn:aws:kms:us-west-2:111122223333:alias/finance-key`.

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/developer"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RequestAlias": "alias/finance-key"
    }
  }
}
```

Vous ne pouvez pas utiliser cette clé de condition pour contrôler l'accès aux opérations d'alias, telles que CreateAliasou DeleteAlias. Pour de plus amples informations sur le contrôle de l'accès aux opérations d'alias, veuillez consulter Contrôle de l'accès aux alias.

km : ResourceAliases

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:ResourceAliases`	Chaîne (liste)	Multi-valeurs	Opérations liées aux ressources de clé KMS	Politiques IAM uniquement

Utilisez cette clé de condition pour contrôler l'accès à une clé KMS en fonction des alias associés à la clé KMS. L'opération doit être une opération de ressource de clé KMS, c'est-à-dire une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS key` dans la colonne `Resources` de l'opération.

Cette condition prend en charge le contrôle d'accès basé sur les attributs (ABAC) dans AWS KMS. Avec l'ABAC, vous pouvez contrôler l'accès aux clés KMS en fonction des balises attribuées à une clé KMS et des alias associés à une clé KMS. Vous pouvez utiliser des balises et des alias pour autoriser ou refuser l'accès à une clé KMS sans modifier les politiques ou les octrois. Pour plus de détails, veuillez consulter ABAC pour AWS KMS.

Un alias doit être unique dans une région Compte AWS et, mais cette condition vous permet de contrôler l'accès à plusieurs clés KMS dans la même région (à l'aide de l'opérateur Régions AWS de `StringLike` comparaison) ou à plusieurs clés KMS dans différents comptes.

> **Note**
>
> La ResourceAliases condition kms : n'est effective que lorsque la clé KMS est conforme aux alias par quota de clé KMS. Si une clé KMS dépasse ce quota, les principaux autorisés à utiliser la clé KMS par la condition `kms:ResourceAliases` se voient refuser l'accès à la clé KMS.

Pour spécifier l'alias dans cette condition de politique, utilisez un nom d'alias, tel que `alias/project-alpha`, ou un modèle de nom d'alias, tel que `alias/*test*`. Vous ne pouvez pas spécifier un ARN d'alias dans la valeur de cette clé de condition. Pour satisfaire à cette condition, la

clé KMS utilisée dans l'opération doit avoir l'alias spécifié. Peu importe si ou comment la clé KMS est identifiée dans la demande pour l'opération.

Il s'agit d'une clé de condition multi-valeurs qui compare l'ensemble d'alias associé à une clé KMS à l'ensemble d'alias de la politique. Pour déterminer comment ces ensembles sont comparés, vous devez fournir un opérateur d'ensemble `ForAnyValue` ou `ForAllValues` dans la condition de politique. Pour plus de détails sur les opérateurs d'ensemble, veuillez consulter Utilisation de plusieurs clés et valeurs dans le Guide de l'utilisateur IAM.

- ForAnyValue: Au moins un alias associé à la clé KMS doit correspondre à un alias figurant dans la condition de politique. D'autres alias sont autorisés. Si la clé KMS n'a pas d'alias, la condition n'est pas remplie.
- ForAllValues: Chaque alias associé à la clé KMS doit correspondre à un alias indiqué dans la politique. Cet opérateur d'ensemble limite les alias associés à la clé KMS à ceux de la condition de politique. Il ne nécessite aucun alias, mais il interdit les alias non spécifiés.

Par exemple, la déclaration de politique IAM suivante permet au principal d'appeler l'GenerateDataKeyopération sur n'importe quelle clé KMS spécifiée Compte AWS associée à l'`finance-key`alias. (Les politiques de clé des clés KMS affectées doivent également permettre au compte du principal de les utiliser pour cette opération.) Pour indiquer que la condition est remplie lorsque l'un des nombreux alias qui peuvent être associés à la clé KMS est `alias/finance-key`, la condition utilise l'opérateur d'ensemble `ForAnyValue`.

Puisque la condition `kms:ResourceAliases` est basée sur la ressource et non pas sur la demande, un appel vers `GenerateDataKey` réussit pour toute clé KMS associée à l'alias `finance-key`, même si la demande utilise un ID de clé ou un ARN de clé pour identifier la clé KMS.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": "kms:GenerateDataKey",
  "Resource": [
    "arn:aws:kms:*:111122223333:key/*",
    "arn:aws:kms:*:444455556666:key/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:ResourceAliases": "alias/finance-key"
    }
```

```
  }
}
```

L'exemple suivant de politique IAM permet au principal d'activer et de désactiver les clés KMS, mais uniquement lorsque tous les alias des clés KMS incluent « `Test` ». Cette instruction de politique utilise deux conditions. La condition avec l'opérateur d'ensemble `ForAllValues` exige que tous les alias associés à la clé KMS incluent « Test ». La condition avec l'opérateur d'ensemble `ForAnyValue` exige que la clé KMS ait au moins un alias avec « Test ». Sans la condition `ForAnyValue`, cette instruction de politique aurait autorisé le principal à utiliser les clés KMS sans alias.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    },
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    }
  }
}
```

km : ReplicaRegion

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:ReplicaRegion`	Chaîne (liste)	À valeur unique	`ReplicateKey`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour limiter la Régions AWS capacité d'un principal à répliquer une clé multirégionale. La clé de `kms:ReplicaRegion` condition contrôle l'accès à l'ReplicateKeyopération en fonction de la valeur du ReplicaRegionparamètre dans la demande. Ce paramètre spécifie la Région AWS pour la nouvelle clé de réplica.

La valeur de la condition est un ou plusieurs Région AWS noms, tels que `us-east-1` ou`ap-southeast-2`, ou des modèles de noms, tels que`eu-*`. Pour obtenir la liste des noms de Régions AWS ces AWS KMS supports, consultez la section AWS Key Management Service Points de terminaison et quotas dans le Références générales AWS.

Par exemple, la déclaration de politique clé suivante utilise la clé de `kms:ReplicaRegion` condition pour permettre aux principaux d'appeler l'ReplicateKeyopération uniquement lorsque la valeur du `ReplicaRegion` paramètre est l'une des régions spécifiées.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey"
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
         "us-east-1",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

Cette clé de condition contrôle uniquement l'accès à l'ReplicateKeyopération. Pour contrôler l'accès à l'UpdatePrimaryRegionopération, utilisez la clé de PrimaryRegion condition kms :.

km : RetiringPrincipal

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:RetiringPrincipal`	Chaîne (liste)	À valeur unique	`CreateGrant`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour contrôler l'accès à l'CreateGrantopération en fonction de la valeur du RetiringPrincipalparamètre dans la demande. Par exemple, vous pouvez créer des octrois d'utilisation d'une clé KMS uniquement lorsque le `RetiringPrincipal` de la demande `CreateGrant` correspond au `RetiringPrincipal` spécifié dans l'instruction de condition.

Pour spécifier le principal sortant, utilisez le Amazon Resource Name (ARN) d'un AWS principal. Les principaux valides incluent les utilisateurs IAM Comptes AWS, les rôles IAM, les utilisateurs fédérés et les utilisateurs des rôles assumés. Pour obtenir de l'aide sur la syntaxe ARN d'un principal, consultez la section ARN IAM dans le guide de l'utilisateur IAM.

L'exemple de déclaration de politique clé suivant permet à un utilisateur de créer des autorisations pour la clé KMS. La clé de `kms:RetiringPrincipal` condition restreint l'autorisation aux `CreateGrant` demandes pour lesquelles le principal sortant de la subvention est le. `LimitedAdminRole`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RetiringPrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

Voir aussi

- km : GrantConstraintType
- km : GrantIsFor AWSResource
- km : GrantOperations
- km : GranteePrincipal

km : ScheduleKeyDeletionPendingWindowInDays

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:ScheduleKeyDeletionPendingWindowInDays`	Numérique	À valeur unique	`ScheduleKeyDeletion`	Politiques de clé et politiques IAM

Vous pouvez utiliser cette clé de condition pour limiter les valeurs que les principaux peuvent spécifier dans le `PendingWindowInDays` paramètre d'une ScheduleKeyDeletiondemande.

`PendingWindowInDays`Spécifie le nombre de jours à AWS KMS attendre avant de supprimer une clé. AWS KMS vous permet de spécifier une période d'attente comprise entre 7 et 30 jours, mais vous pouvez utiliser la clé de `kms:ScheduleKeyDeletionPendingWindowInDays` condition pour limiter davantage la période d'attente, par exemple en imposant une période d'attente minimale comprise dans la plage valide.

Par exemple, l'instruction de politique de clé suivante utilise la clé de condition `kms:ScheduleKeyDeletionPendingWindowInDays` pour empêcher les principaux de planifier la suppression des clés si le délai d'attente est inférieur ou égal à 21 jours.

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
```

```
        "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
      }
  }
}
```

km : SigningAlgorithm

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:SigningAlgorithm`	Chaîne	À valeur unique	`Sign` `Verify`	Politiques de clé et politiques IAM

Vous pouvez utiliser la clé de `kms:SigningAlgorithm` condition pour contrôler l'accès aux opérations de signature et de vérification en fonction de la valeur du SigningAlgorithmparamètre dans la demande. Cette clé de condition n'a aucun effet sur les opérations effectuées en dehors de AWS KMS, telles que la vérification des signatures avec la clé publique dans une paire de clés KMS asymétrique en dehors de AWS KMS.

L'exemple de politique de clé suivant permet aux utilisateurs pouvant endosser le rôle `testers` d'utiliser la clé KMS pour signer des messages uniquement lorsque l'algorithme de signature utilisé pour la demande est un algorithme RSASSA_PSS, tel que `RSASSA_PSS_SHA512`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:SigningAlgorithm": "RSASSA_PSS*"
    }
  }
}
```

Voir aussi

- km : EncryptionAlgorithm
- the section called “km : MacAlgorithm”
- the section called “km : MessageType”

km : ValidTo

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d’API	Type de stratégie
`kms:ValidTo`	Horodatage	À valeur unique	`ImportKeyMaterial`	Politiques de clé et politiques IAM

La clé de `kms:ValidTo` condition contrôle l'accès à l'ImportKeyMaterialopération en fonction de la valeur du ValidToparamètre dans la demande, qui détermine la date d'expiration du matériel clé importé. La valeur est exprimée en heure Unix.

Par défaut, le paramètre `ValidTo` est obligatoire dans une demande `ImportKeyMaterial`. Toutefois, si la valeur du ExpirationModelparamètre est`KEY_MATERIAL_DOES_NOT_EXPIRE`, le `ValidTo` paramètre n'est pas valide. Vous pouvez également utiliser la clé de ExpirationModel condition kms : pour demander le `ExpirationModel` paramètre ou une valeur de paramètre spécifique.

L'exemple d'instruction de politique suivant autorise un utilisateur à importer des éléments de clé dans une clé KMS. La clé de condition `kms:ValidTo` limite l'autorisation aux requêtes `ImportKeyMaterial` où la valeur `ValidTo` est inférieure ou égale à `1546257599.0` (31 décembre 2018 11:59:59 PM).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:ValidTo": "1546257599.0"
```

```
    }
  }
}
```

Voir aussi

- km : ExpirationModel
- km : WrappingAlgorithm
- km : WrappingKeySpec

km : ViaService

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:ViaService`	Chaîne	À valeur unique	Opérations liées aux ressources de clé KMS	Politiques de clé et politiques IAM

La clé de `kms:ViaService` condition limite l'utilisation d'une clé KMS aux demandes provenant de AWS services spécifiques. Vous pouvez spécifier un ou plusieurs services dans chaque clé de condition `kms:ViaService`. L'opération doit être une opération de ressource de clé KMS, c'est-à-dire une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table Actions and Resources (Actions et ressources), recherchez la valeur de `KMS key` dans la colonne `Resources` de l'opération.

Par exemple, l'instruction de politique de clé suivante utilise la clé de condition `kms:ViaService` pour autoriser l'utilisation d'une clé gérée par le client pour les actions spécifiées uniquement lorsque la demande provient d'Amazon EC2 ou Amazon RDS dans la région USA Ouest (Oregon), au nom de `ExampleRole`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
```

```
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "ec2.us-west-2.amazonaws.com",
        "rds.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

Vous pouvez également utiliser une clé de condition `kms:ViaService` pour refuser l'autorisation d'utiliser une clé KMS lorsque la demande provient de services particuliers. Par exemple, l'instruction suivante de politique d'une politique de clé utilise une clé de condition `kms:ViaService` pour empêcher une clé gérée par le client d'être utilisée pour les opérations `Encrypt` lorsque la demande provient de AWS Lambda au nom de `ExampleRole`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
          "lambda.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

> **Important**
>
> Lorsque vous utilisez la clé de condition `kms:ViaService`, le service effectue la demande au nom d'un principal dans le Compte AWS. Ces principaux doivent disposer des autorisations suivantes :
>
> - Autorisation d'utiliser la clé KMS. Le principal doit accorder ces autorisations au service intégré pour que celui-ci puisse utiliser la clé gérée par le client au nom du principal. Pour de plus amples informations, veuillez consulter Comment les services AWS utilisent AWS KMS.
> - L'autorisation d'utiliser le service intégré. Pour en savoir plus sur l'accès des utilisateurs à un AWS service intégré AWS KMS, consultez la documentation du service intégré.

Toutes les Clés gérées par AWS utilisent une clé de condition `kms:ViaService` dans leur document de politique de clé. Cette condition permet à la clé KMS d'être utilisée uniquement pour les demandes qui proviennent du service qui a créé la clé KMS. Pour voir la politique clé d'un Clé gérée par AWS, utilisez l'GetKeyPolicyopération.

La clé de condition `kms:ViaService` est valide dans les instructions de la politique IAM et de la politique de clé. Les services que vous spécifiez doivent être intégrés à AWS KMS et prendre en charge la clé de condition `kms:ViaService`.

Services prenant en charge la clé de condition **`kms:ViaService`**

Le tableau suivant répertorie les AWS services intégrés à la clé de `kms:ViaService` condition dans les clés gérées par le client AWS KMS et prenant en charge son utilisation. Les services de ce tableau peuvent ne pas être disponibles dans toutes les régions. Utilisez le `.amazonaws.com` suffixe du AWS KMS ViaService nom dans toutes les AWS partitions.

> **Note**
>
> Il peut être nécessaire de faire défiler horizontalement ou verticalement pour afficher toutes les données de ce tableau.

Nom du service	AWS KMS ViaService nom
AWS App Runner	apprunner. *AWS_region* .amazonaws.com
AWS AppFabric	appfabric. *AWS_region* .amazonaws.com
Amazon AppFlow	appflow.*AWS_region* .amazonaws.com
AWS Application Migration Service	mgn.*AWS_region* .amazonaws.com
Amazon Athena	athena.*AWS_region* .amazonaws.com
AWS Audit Manager	auditmanager. *AWS_region* .amazonaws.com
Amazon Aurora	rds.*AWS_region* .amazonaws.com
AWS Backup	backup.*AWS_region* .amazonaws.com
PasserelleAWS Backup	backup-gateway. *AWS_region* .amazonaws.com
Kit SDK Amazon Chime	chimevoiceconnector. *AWS_region* .amazonaws.com
AWS CodeArtifact	codeartifact. *AWS_region* .amazonaws.com
CodeGuru Réviseur Amazon	codeguru-reviewer. *AWS_region* .amazonaws.com
Amazon Comprehend	comprehend. *AWS_region* .amazonaws.com
Amazon Connect	connect.*AWS_region* .amazonaws.com
Profils des clients Amazon Connect	profile.*AWS_region* .amazonaws.com

Nom du service	AWS KMS ViaService nom
Amazon Q in Connect	`wisdom.`*`AWS_region`*` .amazonaws.com`
AWS Database Migration Service (AWS DMS)	`dms.`*`AWS_region`*` .amazonaws.com`
AWS Directory Service	`directoryservice.`*`AWS_regio n`*` .amazonaws.com`
Amazon DynamoDB	`dynamodb.`*`AWS_region`*` .amazonaw s.com`
Amazon DocumentDB	`docdb-elastic.`*`AWS_region`*` .amazonaw s.com`
Amazon EC2 Systems Manager (SSM)	`ssm.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic Block Store (Amazon EBS)	`ec2.`*`AWS_region`*` .amazonaws.com` (EBS uniquement)
Amazon Elastic Container Registry (Amazon ECR)	`ecr.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic File System (Amazon EFS)	`elasticfilesystem.`*`AWS_regio n`*` .amazonaws.com`
Amazon ElastiCache	Incluez les deux ViaService noms dans la valeur de la clé de condition : • `elasticache.`*`AWS_region`*` .amazonaw s.com` • `dax.`*`AWS_region`*` .amazonaws.com`
AWS Elemental MediaTailor	`mediatailor.`*`AWS_region`*` .amazonaw s.com`
AWS Résolution de l'entité	`entityresolution.`*`AWS_regio n`*` .amazonaws.com`

Nom du service	AWS KMS ViaService nom
Amazon FinSpace	finspace. *AWS_region* .amazonaws.com
Amazon Forecast	forecast. *AWS_region* .amazonaws.com
Amazon FSx	fsx.*AWS_region* .amazonaws.com
AWS Glue	glue.*AWS_region* .amazonaws.com
AWS Ground Station	groundstation. *AWS_region* .amazonaws.com
Amazon GuardDuty	malware-protection. *AWS_region* .amazonaws.com
AWS HealthLake	healthlake. *AWS_region* .amazonaws.com
AWS IoT SiteWise	iotsitewise. *AWS_region* .amazonaws.com
Amazon Kendra	kendra.*AWS_region* .amazonaws.com
Amazon Keyspaces (pour Apache Cassandra)	cassandra. *AWS_region* .amazonaws.com
Amazon Kinesis	kinesis.*AWS_region* .amazonaws.com
Amazon Data Firehose	firehose. *AWS_region* .amazonaws.com
Amazon Kinesis Video Streams	kinesisvideo. *AWS_region* .amazonaws.com
AWS Lambda	lambda.*AWS_region* .amazonaws.com
Amazon Lex	lex.*AWS_region* .amazonaws.com

Nom du service	AWS KMS ViaService nom
AWS License Manager	`license-manager.` *`AWS_region`* `.amazonaws.com`
Amazon Location Service	`geo.`*`AWS_region`* `.amazonaws.com`
Amazon Lookout for Equipment	`lookoutequipment.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout for Metrics	`lookoutmetrics.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout for Vision	`lookoutvision.` *`AWS_region`* `.amazonaws.com`
Amazon Macie	`macie.`*`AWS_region`* `.amazonaws.com`
AWS Mainframe Modernization	`m2.`*`AWS_region`* `.amazonaws.com`
Amazon Managed Blockchain	`managedblockchain.` *`AWS_region`* `.amazonaws.com`
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	`kafka.`*`AWS_region`* `.amazonaws.com`
Amazon Managed Workflows for Apache Airflow (MWAA)	`airflow.`*`AWS_region`* `.amazonaws.com`
Amazon MemoryDB for Redis	`memorydb.` *`AWS_region`* `.amazonaws.com`
Amazon Monitron	`monitron.` *`AWS_region`* `.amazonaws.com`
Amazon MQ	`mq.`*`AWS_region`* `.amazonaws.com`
Amazon Neptune	`rds.`*`AWS_region`* `.amazonaws.com`
Amazon Nimble Studio	`nimble.`*`AWS_region`* `.amazonaws.com`

Nom du service	AWS KMS ViaService nom
AWS HealthOmics	omics.*AWS_region* .amazonaws.com
Amazon OpenSearch Service	es.*AWS_region* .amazonaws.com , aoss.*AWS_region* .amazonaws.com
AWS Proton	proton.*AWS_region* .amazonaws.com
Amazon Quantum Ledger Database (Amazon QLDB)	qldb.*AWS_region* .amazonaws.com
Analyse des performances d'Amazon RDS	rds.*AWS_region* .amazonaws.com
Amazon Redshift	redshift. *AWS_region* .amazonaws.com
Ouvrez l'éditeur de requête V2 Amazon Redshift.	sqlworkbench. *AWS_region* .amazonaws.com
Amazon Redshift Serverless	redshift-serverless. *AWS_region* .amazonaws.com
Amazon Rekognition	rekognition. *AWS_region* .amazonaws.com
Amazon Relational Database Service (Amazon RDS)	rds.*AWS_region* .amazonaws.com
Stockage de données répliquées Amazon	ards.*AWS_region* .amazonaws.com
Amazon SageMaker	sagemaker. *AWS_region* .amazonaws.com
AWS Secrets Manager	secretsmanager. *AWS_region* .amazonaws.com
Amazon Security Lake	securitylake. *AWS_region* .amazonaws.com

Nom du service	AWS KMS ViaService nom
Amazon Simple Email Service (Amazon SES)	ses.*AWS_region* .amazonaws.com
Amazon Simple Notification Service (Amazon SNS)	sns.*AWS_region* .amazonaws.com
Amazon Simple Queue Service (Amazon SQS)	sqs.*AWS_region* .amazonaws.com
Amazon Simple Storage Service (Amazon S3)	s3.*AWS_region* .amazonaws.com
AWS Snowball	importexport. *AWS_region* .amazonaws.com
AWS Storage Gateway	storagegateway. *AWS_region* .amazonaws.com
AWS Systems Manager Incident Manager	ssm-incidents. *AWS_region* .amazonaws.com
AWS Systems Manager Incident Manager Contacts	ssm-contacts. *AWS_region* .amazonaws.com
Amazon Timestream	timestream. *AWS_region* .amazonaws.com
Amazon Translate	translate. *AWS_region* .amazonaws.com
Accès vérifié par AWS	verified-access. *AWS_region* .amazonaws.com
Amazon WorkMail	workmail. *AWS_region* .amazonaws.com
Amazon WorkSpaces	workspaces. *AWS_region* .amazonaws.com
Amazon WorkSpaces Thin Client	thinclient. *AWS_region* .amazonaws.com

Nom du service	AWS KMS ViaService nom
WorkSpaces Site Web d'Amazon	`workspaces-web.` *`AWS_region`* `.amazonaws.com`
AWS X-Ray	`xray.`*`AWS_region`*` .amazonaws.com`

km : WrappingAlgorithm

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:WrappingAlgorithm`	Chaîne	À valeur unique	`GetParametersForImport`	Politiques de clé et politiques IAM

Cette clé de condition contrôle l'accès à l'GetParametersForImportopération en fonction de la valeur du WrappingAlgorithmparamètre dans la demande. Vous pouvez utiliser cette condition pour exiger des principaux qu'ils utilisent un algorithme particulier pour chiffrer des matériaux clé au cours du processus d'importation. Les demandes de clé publique requise et du jeton d'importation échouent lorsqu'un algorithme d'encapsulage est spécifié.

L'exemple d'instruction de politique de clé suivant utilise la clé de condition `kms:WrappingAlgorithm` pour donner à l'utilisateur de l'exemple l'autorisation d'appeler l'opération `GetParametersForImport`, mais l'empêche d'utiliser l'algorithme d'encapsulage `RSAES_OAEP_SHA_1`. Lorsque `WrappingAlgorithm` dans la requête `GetParametersForImport` indique `RSAES_OAEP_SHA_1`, l'opération échoue.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
```

```
    }
  }
}
```

Voir aussi

- km : ExpirationModel
- km : ValidTo
- km : WrappingKeySpec

km : WrappingKeySpec

AWS KMS clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de stratégie
`kms:WrappingKeySpec`	Chaîne	À valeur unique	`GetParametersForImport`	Politiques de clé et politiques IAM

Cette clé de condition contrôle l'accès à l'GetParametersForImportopération en fonction de la valeur du WrappingKeySpecparamètre dans la demande. Vous pouvez utiliser cette condition pour exiger des principaux qu'ils utilisent un type particulier de clé publique au cours du processus d'importation. Si la requête spécifie un autre type de clé, elle échoue.

Étant donné que la seule valeur valide comme valeur du paramètre `WrappingKeySpec` est `RSA_2048`, les utilisateurs ne peuvent pas employer cette valeur de façon efficace, ce qui les empêche d'utiliser l'opération `GetParametersForImport`.

L'exemple d'instruction de politique suivant utilise la clé de condition `kms:WrappingAlgorithm` pour exiger que la valeur du paramètre `WrappingKeySpec` de la demande soit `RSA_4096`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
```

```
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```

Voir aussi

- km : ExpirationModel
- km : ValidTo
- km : WrappingAlgorithm

AWS KMS clés de condition pour AWS Nitro Enclaves

AWS Nitro Enclaves est une fonctionnalité Amazon EC2 qui vous permet de créer des environnements informatiques isolés appelés enclaves pour protéger et traiter des données hautement sensibles. AWS KMS fournit des clés de condition pour prendre en charge AWS Nitro Enclaves. Ces clés de conditions ne sont efficaces que pour les demandes adressées à AWS KMS une Nitro Enclave.

Lorsque vous appelez les opérations Decrypt, GenerateDataKeyGenerateDataKeyPair, ou GenerateRandomAPI avec le document d'attestation signé depuis une enclave, ces API chiffrent le texte en clair de la réponse sous la clé publique du document d'attestation et renvoient du texte chiffré au lieu du texte en clair. Ce texte chiffré peut être déchiffré uniquement à l'aide de la clé privée dans l'enclave. Pour de plus amples informations, veuillez consulter Comment AWS Nitro Enclaves utilise AWS KMS.

Les clés de condition suivantes vous permettent de limiter les autorisations pour ces opérations en fonction du contenu du document d'attestation signé. Avant d'autoriser une opération, AWS KMS compare le document d'attestation de l'enclave aux valeurs de ces clés de AWS KMS condition.

km RecipientAttestation : ImageSha 384

AWS KMS Clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de politique
`kms:RecipientAttes`	Chaîne	À valeur unique	`Decrypt`	Politiques de clé et politiques IAM

AWS KMS Clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de politique

`tation:Im ageSha384`			`GenerateD ataKey`
			
			`GenerateD ataKeyPair`
			
			`GenerateR andom`	

La clé de condition `kms:RecipientAttestation:ImageSha384` contrôle l'accès à `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair` et `GenerateRandom` avec une clé KMS lorsque le résumé d'image du document d'attestation signé dans la demande correspond à la valeur de la clé de condition. La valeur `ImageSha384` correspond au PCR0 du document d'attestation. Cette clé de condition n'est effective que lorsque le `Recipient` paramètre de la demande spécifie un document d'attestation signé pour une enclave AWS Nitro.

Cette valeur est également incluse dans les CloudTrailévénements relatifs aux demandes adressées AWS KMS aux enclaves Nitro.

> **Note**
>
> Cette clé de condition est valide dans les instructions de politique de clé et les instructions de politique IAM, même si elle n'apparaît pas dans la console IAM ou dans la référence d'autorisations de service IAM.

Par exemple, la déclaration de politique clé suivante autorise le `data-processing` rôle à utiliser la clé KMS pour les GenerateRandomopérations de déchiffrement GenerateDataKey, GenerateDataKeyPair, et. La clé de condition `kms:RecipientAttestation:ImageSha384` permet les opérations uniquement lorsque la valeur de hachage de l'image (PCR0) du document d'attestation de la demande correspond à la valeur de hachage de l'image de la condition. Cette clé de condition n'est effective que lorsque le `Recipient` paramètre de la demande spécifie un document d'attestation signé pour une enclave AWS Nitro.

Si la demande n'inclut pas de document d'attestation valide provenant d'une enclave AWS Nitro, l'autorisation est refusée car cette condition n'est pas remplie.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

km : : PCR RecipientAttestation <PCR_ID>

AWS KMS Clés de condition	Type de condition	Type de la valeur	Opérations d'API	Type de politique

`kms:RecipientAttestation:PCR<PCR_ID>`	Chaîne	À valeur unique	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Politiques de clé et politiques IAM

La clé de condition `kms:RecipientAttestation:PCR<PCR_ID>` contrôle l'accès à `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair` et `GenerateRandom` avec une clé KMS uniquement lorsque les registres de configuration de plateforme (PCR) du document d'attestation signé de la demande correspondent aux PCR de la clé de condition. Cette clé de condition n'est effective que lorsque le `Recipient` paramètre de la demande spécifie un document d'attestation signé provenant d'une enclave AWS Nitro.

Cette valeur est également incluse dans les CloudTrailévénements qui représentent des demandes adressées à AWS KMS des enclaves Nitro.

> **Note**
>
> Cette clé de condition est valide dans les instructions de politique de clé et les instructions de politique IAM, même si elle n'apparaît pas dans la console IAM ou dans la référence d'autorisations de service IAM.

Pour spécifier une valeur PCR, utilisez le format suivant. Concaténez l'ID de PCR au nom de clé de condition. La valeur PCR doit être une chaîne hexadécimale en minuscules de 96 octets maximum.

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```

Par exemple, la clé de condition suivante spécifie une valeur particulière pour PCR1, qui correspond au hachage du noyau utilisé pour l'enclave et le processus d'amorçage.

```
kms:RecipientAttestation:PCR1:
 "0x1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcdef
```

Par exemple, l'instruction de stratégie de clé suivante autorise le rôle `data-processing` à utiliser la clé KMS pour l'opération Decrypt.

La clé de condition `kms:RecipientAttestation:PCR` de cette instruction autorise l'opération uniquement lorsque la valeur PCR1 du document d'attestation signé dans la demande correspond à la valeur `kms:RecipientAttestation:PCR1` de la condition. Utilisez l'opérateur de politique `StringEqualsIgnoreCase` pour exiger une comparaison insensible à la casse des valeurs PCR.

Si la demande n'inclut pas de document d'attestation, l'autorisation est refusée car cette condition n'est pas remplie.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
 "0x1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f15
    }
  }
}
```

ABAC pour AWS KMS

Le contrôle d'accès basé sur les attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. AWS KMS prend en charge l'ABAC en vous permettant de contrôler l'accès à vos clés gérées par le client en fonction des balises et alias associés aux clés KMS. Les clés de condition de balise et d'alias qui activent l'ABAC dans AWS KMS offrent un moyen puissant et flexible d'autoriser les principaux à utiliser les clés KMS sans modifier les politiques ou gérer les octrois. Toutefois, vous devriez utiliser cette fonction avec précaution, afin que les principaux ne soient pas autorisés ou refusés par inadvertance.

Si vous utilisez l'ABAC, sachez que l'autorisation de gérer les balises et les alias est désormais une autorisation de contrôle d'accès. Assurez-vous de connaître les balises et alias existants sur toutes les clés KMS avant de déployer une politique qui en dépend. Prenez des précautions raisonnables lors de l'ajout, de la suppression et de la mise à jour des alias, ainsi que lors de l'étiquetage et du désétiquetage des clés. Accordez des autorisations pour gérer les balises et les alias uniquement aux principaux qui en ont besoin, et limitez les balises et les alias qu'ils peuvent gérer.

> **Remarques**
>
> Lors de l'utilisation d'ABAC pour AWS KMS, soyez prudent lorsque vous autorisez les principaux à gérer les balises et les alias. La modification d'une balise ou d'un alias peut autoriser ou refuser l'accès à une clé KMS. Les administrateurs de clés qui n'ont pas

l'autorisation de modifier les politiques de clé ou de créer des octrois peuvent contrôler l'accès aux clés KMS s'ils sont autorisés à gérer les balises ou les alias.
Les modifications d'alias et de balises peuvent prendre jusqu'à cinq minutes pour affecter l'autorisation de clé KMS. Les modifications récentes peuvent être visibles dans les opérations d'API avant qu'elles n'affectent l'autorisation.
Pour contrôler l'accès à une clé KMS en fonction de son alias, vous devez utiliser une clé de condition. Vous ne pouvez pas utiliser un alias pour représenter une clé KMS dans l'élément `Resource` d'une instruction de politique. Lorsqu'un alias apparaît dans l'élément `Resource`, l'instruction de politique s'applique à l'alias et non à la clé KMS associée.

En savoir plus

- Pour plus de détails sur la prise en charge de AWS KMS pour l'ABAC, y compris des exemples, veuillez consulter Utilisation d'alias pour contrôler l'accès aux clés KMS et Utilisation de balises pour contrôler l'accès aux clés KMS.
- Pour plus d'informations générales sur l'utilisation des balises pour contrôler l'accès aux ressources AWS, veuillez consulter Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) pour AWS ? et Contrôle de l'accès aux ressources AWS à l'aide de balises dans le Guide de l'utilisateur IAM.

Clés de condition ABAC pour AWS KMS

Pour autoriser l'accès aux clés KMS en fonction de leurs balises et alias, utilisez les clés de condition suivantes dans une politique de clé ou une politique IAM.

Clé de condition ABAC	Description	Type de stratégie	Opérations AWS KMS
lois : ResourceTag	La balise (clé et valeur) de la clé KMS correspond à la balise (clé et valeur) ou au modèle de balise dans la politique.	Politique IAM uniquement	Opérations liées aux ressources de clé KMS [2]

Clé de condition ABAC	Description	Type de stratégie	Opérations AWS KMS
aws :RequestTag//tag-key	La balise (clé et valeur) dans la demande correspond à la balise (clé et valeur) ou au modèle de balise dans la politique.	Politique de clé et politiques IAM[1]	TagResource, UntagResource
lois : TagKeys	Dans la demande, les clés de balise correspondent à celles de la politique.	Politique de clé et politiques IAM[1]	TagResource, UntagResource
km : ResourceAliases	Les alias associés à la clé KMS correspondent aux alias ou aux modèles d'alias de la politique.	Politique IAM uniquement	Opérations liées aux ressources de clé KMS [2]
km : RequestAlias	L'alias qui représente la clé KMS dans la demande correspond à l'alias ou aux modèles d'alias de la politique.	Politique de clé et politiques IAM[1]	opérations cryptographiques, DescribeKeyGetPublicKey

[1]Toute clé de condition pouvant être utilisée dans une politique de clé peut également être utilisée dans une politique IAM, mais uniquement si la politique clé le permet.

[2]Une opération de ressource de clé KMS est une opération autorisée pour une clé KMS particulière. Pour identifier les opérations de ressources de clés KMS, dans la table AWS KMS des autorisations, recherchez la valeur de la clé KMS dans la colonne `Resources` de l'opération.

Par exemple, vous pouvez utiliser ces clés de condition pour créer les politiques suivantes.

- Une politique IAM avec `kms:ResourceAliases` qui autorise l'utilisation de clés KMS avec un alias ou un modèle d'alias particulier. Cela est un peu différent des politiques qui reposent sur des balises : bien que vous puissiez utiliser des modèles d'alias dans une politique, chaque alias doit être unique dans un Compte AWS et une région. Cela vous permet d'appliquer une politique à un ensemble de clés KMS sélectionné sans répertorier les ARN des clés KMS dans l'instruction de politique. Pour ajouter ou supprimer des clés KMS de l'ensemble, modifiez l'alias de la clé KMS.
- Une politique de clé avec `kms:RequestAlias` qui permet aux principaux d'utiliser une clé KMS dans une opération `Encrypt`, mais uniquement lorsque la demande `Encrypt` utilise cet alias pour identifier la clé KMS.
- Une politique IAM avec `aws:ResourceTag/`*`tag-key`* qui refuse l'autorisation d'utiliser des clés KMS avec une clé et une valeur de balise particulières. Cela vous permet d'appliquer une politique à un ensemble de clés KMS sélectionné sans répertorier les ARN des clés KMS dans l'instruction de politique. Pour ajouter ou supprimer des clés KMS de l'ensemble, étiquetez ou désétiquetez la clé KMS.
- Une politique IAM avec `aws:RequestTag/`*`tag-key`* qui permet aux principaux de supprimer uniquement les balises de clés KMS `"Purpose"="Test"`.
- Une politique IAM avec `aws:TagKeys` qui refuse l'autorisation d'étiqueter ou de désétiqueter une clé KMS avec une clé de balise `Restricted`.

L'ABAC rend la gestion des accès flexible et évolutive. Par exemple, vous pouvez utiliser la clé de condition `aws:ResourceTag/`*`tag-key`* pour créer une politique IAM qui permet aux principaux d'utiliser une clé KMS pour des opérations spécifiées uniquement lorsque la clé KMS possède une balise `Purpose=Test`. La politique s'applique à toutes les clés KMS dans toutes les régions du Compte AWS.

Lorsqu'elle est attachée à un utilisateur ou à un rôle, la politique IAM suivante permet aux principaux d'utiliser toutes les clés KMS existantes avec une balise `Purpose=Test` pour les opérations spécifiées. Pour autoriser cet accès à des clés KMS nouvelles ou existantes, vous n'avez pas besoin de modifier la politique. Il suffit de joindre la balise `Purpose=Test` aux clés KMS. De même, pour supprimer cet accès des clés KMS avec une balise `Purpose=Test`, modifiez ou supprimez la balise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
```

```
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

Toutefois, si vous utilisez cette fonction, faites attention lors de la gestion des balises et des alias. L'ajout, la modification ou la suppression d'une balise ou d'un alias peut autoriser ou refuser l'accès à une clé KMS par inadvertance. Les administrateurs de clés qui n'ont pas l'autorisation de modifier les politiques de clé ou de créer des octrois peuvent contrôler l'accès aux clés KMS s'ils sont autorisés à gérer les balises et les alias. Pour atténuer ce risque, envisagez de limiter les autorisations de gestion des balises et des alias. Par exemple, vous pouvez autoriser uniquement les principaux sélectionnés à gérer les balises `Purpose=Test`. Pour plus de détails, veuillez consulter Utilisation d'alias pour contrôler l'accès aux clés KMS et Utilisation de balises pour contrôler l'accès aux clés KMS.

Des balises ou des alias ?

AWS KMS prend en charge l'ABAC avec des balises et des alias. Les deux options offrent une stratégie de contrôle d'accès flexible et évolutive, mais elles sont légèrement différentes l'une de l'autre.

Vous pouvez décider d'utiliser des balises ou des alias en fonction de vos modèles d'utilisation AWS particuliers. Par exemple, si vous avez déjà accordé des autorisations d'étiquetage à la plupart des administrateurs, il peut être plus facile de contrôler une stratégie d'autorisation basée sur des alias. Ou, si vous approchez du quota pour le nombre d'alias par clé KMS, vous pouvez préférer une stratégie d'autorisation basée sur des balises.

Les avantages suivants sont d'intérêt général.

Avantages du contrôle d'accès basé sur les identifications

- Même mécanisme d'autorisation pour différents types de ressources AWS.

  Vous pouvez utiliser la même balise ou clé de balise pour contrôler l'accès à plusieurs types de ressources, tels qu'un cluster Amazon Relational Database Service (Amazon RDS), un volume Amazon Elastic Block Store (Amazon EBS) et une clé KMS. Cette fonction permet plusieurs modèles d'autorisation plus flexibles que le contrôle d'accès classique basé sur les rôles.

- Autoriser l'accès à un groupe de clés KMS.

  Vous pouvez utiliser des balises pour gérer l'accès à un groupe de clés KMS dans le même Compte AWS et la même région. Attribuez la même balise ou la même clé de balise aux clés KMS que vous choisissez. Créez ensuite une déclaration de easy-to-maintain politique simple basée sur le tag ou la clé du tag. Pour ajouter ou supprimer une clé KMS de votre groupe d'autorisations, ajoutez ou supprimez la balise ; vous n'avez pas besoin de modifier la politique.

Avantages du contrôle d'accès basé sur les alias

- Autoriser l'accès aux opérations cryptographiques en fonction des alias.

  La plupart des conditions de politique basées sur les demandes pour les attributs, y compris aws :RequestTag/tag-key, affectent uniquement les opérations qui ajoutent, modifient ou suppriment l'attribut. Mais la clé de RequestAlias condition kms : contrôle l'accès aux opérations cryptographiques en fonction de l'alias utilisé pour identifier la clé KMS dans la demande. Par exemple, vous pouvez accorder à un principal l'autorisation d'utiliser une clé KMS dans une opération `Encrypt` mais uniquement lorsque la valeur du paramètre `KeyId` est `alias/restricted-key-1`. Cette condition nécessite tous les éléments suivants pour répondre aux exigences :

  - La clé KMS doit être associée à cet alias.
  - La demande doit utiliser l'alias pour identifier la clé KMS.
  - Le principal doit être autorisé à utiliser la clé KMS sujette à la condition `kms:RequestAlias`.

  Cela est particulièrement utile si vos applications utilisent couramment des noms d'alias ou des ARN d'alias pour faire référence à des clés KMS.

- Fournir des autorisations très limitées.

  Un alias doit être unique dans le compte et la région Compte AWS. Par conséquent, donner aux principaux accès à une clé KMS basée sur un alias peut être beaucoup plus restrictif que leur donner un accès basé sur une balise. Contrairement aux alias, les balises peuvent être affectées

à plusieurs clés KMS dans le même compte et la même région. Si vous le souhaitez, vous pouvez utiliser un modèle d'alias, tel que `alias/test*`, pour donner aux principaux accès à un groupe de clés KMS dans le même compte et la même région. Cependant, autoriser ou refuser l'accès à un alias particulier permet un contrôle très strict sur les clés KMS.

Résolution des problèmes liés à l'ABAC pour AWS KMS

Le contrôle de l'accès aux clés KMS en fonction de leurs balises et alias est pratique et puissant. Cependant, cette méthode est sujette à quelques erreurs prévisibles que vous voudrez éviter.

Accès modifié en raison d'un changement de balise

Si une balise est supprimée ou si sa valeur est modifiée, les principaux qui ont accès à une clé KMS basée uniquement sur cette balise se verront refuser l'accès à la clé KMS. Cela peut également se produire lorsqu'une balise incluse dans une instruction de politique de refus est ajoutée à une clé KMS. L'ajout d'une balise liée à une politique à une clé KMS peut permettre l'accès aux principaux qui devraient se voir refuser l'accès à une clé KMS.

Supposons, par exemple, qu'un principal ait accès à une clé KMS basée sur la balise `Project=Alpha`, par exemple l'autorisation fournie par l'exemple d'instruction de politique IAM suivant.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
```

```
}
```

Si la balise est supprimée de cette clé KMS ou si la valeur de la balise est modifiée, le principal n'a plus l'autorisation d'utiliser la clé KMS pour les opérations spécifiées. Cela peut devenir évident lorsque le directeur essaie de lire ou d'écrire des données dans un AWS service qui utilise une clé gérée par le client. Pour suivre le changement de balise, consultez vos CloudTrail journaux TagResourceou UntagResource entrées.

Pour restaurer l'accès sans mettre à jour la politique, modifiez les balises de la clé KMS. Cette mesure a un impact minime sur une brève période et elle prend effet sur l'ensemble de AWS KMS. Pour éviter une erreur comme celle-ci, accordez des autorisations d'étiquetage et de désétiquetage uniquement aux principaux qui en ont besoin et limitez leurs autorisations d'étiquetage aux balises qu'ils doivent gérer. Avant de modifier une balise, recherchez des politiques pour détecter l'accès qui dépend de la balise et obtenir des clés KMS dans toutes les régions qui possèdent la balise. Vous pouvez envisager de créer une CloudWatch alarme Amazon lorsque des balises spécifiques sont modifiées.

Changement d'accès dû à un changement d'alias

Si un alias est supprimé ou associé à une autre clé KMS, les principaux qui ont accès à la clé KMS basée uniquement sur cet alias se verront refuser l'accès à la clé KMS. Cela peut également se produire lorsqu'un alias associé à une clé KMS est inclus dans une instruction de politique de refus. L'ajout d'un alias lié à une politique à une clé KMS peut également permettre l'accès aux principaux qui devraient se voir refuser l'accès à une clé KMS.

Par exemple, la déclaration de politique IAM suivante utilise la clé de ResourceAliases condition kms : pour autoriser l'accès aux clés KMS dans les différentes régions du compte avec l'un des alias spécifiés.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*",
        "kms:Decrypt"
      ],
```

```
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "kms:ResourceAliases": [
            "alias/ProjectAlpha",
            "alias/ProjectAlpha_Test",
            "alias/ProjectAlpha_Dev"
          ]
        }
      }
    }
  ]
}
```

Pour suivre le changement d'alias, consultez vos CloudTrail journaux pour CreateAliasUpdateAlias, et vos DeleteAliasentrées.

Pour restaurer l'accès sans mettre à jour la politique, modifiez les alias associés à la clé KMS. Étant donné que chaque alias ne peut être associé qu'à une seule clé KMS dans un compte et une région, la gestion des alias est un peu plus difficile que la gestion des balises. La restauration de l'accès de certains principaux sur une clé KMS peut refuser au même ou à d'autres principaux l'accès à une autre clé KMS.

Pour éviter cette erreur, n'accordez des autorisations de gestion d'alias qu'aux principaux qui en ont besoin et limitez leurs autorisations de gestion des alias aux alias qu'ils doivent gérer. Avant de mettre à jour ou de supprimer un alias, recherchez des politiques pour détecter l'accès qui dépend de l'alias et recherchez les clés KMS dans toutes les régions associées à l'alias.

Accès refusé en raison d'un quota d'alias

Les utilisateurs autorisés à utiliser une clé KMS dans une limite de kilomètres bénéficieront ResourceAliases d'une `AccessDenied` exception si la clé KMS dépasse les alias par défaut par quota de clé KMS pour ce compte et cette région.

Pour restaurer l'accès, supprimez les alias associés à la clé KMS afin qu'elle soit conforme au quota. Sinon, utilisez un autre mécanisme pour accorder aux utilisateurs l'accès à la clé KMS.

Modification retardée de l'autorisation

Les modifications que vous apportez aux balises et aux alias peuvent prendre jusqu'à cinq minutes pour affecter l'autorisation des clés KMS. Par conséquent, un changement de balise ou d'alias peut être reflété dans les réponses des opérations d'API avant qu'elles n'affectent l'autorisation. Ce délai

est susceptible d'être plus long que le bref délai de cohérence éventuel qui affecte la plupart des opérations AWS KMS.

Par exemple, vous disposez peut-être d'une politique IAM qui autorise certains principaux à utiliser n'importe quelle clé KMS avec une balise `"Purpose"="Test"`. Ensuite, vous ajoutez la balise `"Purpose"="Test"` sur une clé KMS. Bien que l'TagResourceopération soit terminée et que la ListResourceTagsréponse confirme que la balise est attribuée à la clé KMS, les principaux peuvent ne pas avoir accès à la clé KMS pendant cinq minutes au maximum.

Pour éviter les erreurs, intégrez ce délai attendu à votre code.

Demandes ayant échoué en raison des mises à jour d'alias

Lorsque vous mettez à jour un alias, vous associez un alias existant à une autre clé KMS.

Le déchiffrement et les ReEncryptdemandes spécifiant le nom d'alias ou l'ARN de l'alias peuvent échouer car l'alias est désormais associé à une clé KMS qui n'a pas chiffré le texte chiffré. Cette situation renvoie généralement un `IncorrectKeyException` ou `NotFoundException`. Si la demande n'a pas de paramètre `KeyId` ou `DestinationKeyId`, l'opération peut échouer avec l'exception `AccessDenied`, car l'appelant n'a plus accès à la clé KMS qui a chiffré le texte chiffré.

Vous pouvez suivre les modifications en consultant CloudTrail les journaux et CreateAliasUpdateAliasles entrées des DeleteAliasjournaux. Vous pouvez également utiliser la valeur du `LastUpdatedDate` champ dans la ListAliasesréponse pour détecter un changement.

Par exemple, l'ListAliasesexemple de réponse suivant montre que l'`ProjectAlpha_Test`alias de la `kms:ResourceAliases` condition a été mis à jour. Par conséquent, les principaux qui ont un accès en fonction de l'alias perdent leur accès à la clé KMS précédemment associée. Au lieu de cela, ils ont accès à la clé KMS nouvellement associée.

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/ProjectAlpha`)]'

{
    "Aliases": [
        {
            "AliasName": "alias/ProjectAlpha_Test",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ProjectAlpha_Test",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1566518783.394,
            "LastUpdatedDate": 1605308931.903
        },
        {
```

```
            "AliasName": "alias/ProjectAlpha_Restricted",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/
ProjectAlpha_Restricted",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1553410800.010,
            "LastUpdatedDate": 1553410800.010
        }
    ]
}
```

La solution à ce problème n'est pas simple. Vous pouvez à nouveau mettre à jour l'alias pour l'associer à la clé KMS d'origine. Toutefois, avant d'agir, vous devez tenir compte de l'effet de cette modification sur la clé KMS actuellement associée. Si les principaux utilisent cette dernière clé KMS dans des opérations de chiffrement, ils peuvent avoir besoin d'un accès continu à celle-ci. Dans ce cas, vous pouvez mettre à jour la politique pour vous assurer que les principaux ont l'autorisation d'utiliser les deux clés KMS.

Vous pouvez empêcher une erreur comme celle-ci : avant de mettre à jour un alias, examinez les politiques pour détecter l'accès qui dépend de l'alias. Obtenez ensuite les clés KMS dans toutes les régions associées à l'alias. Accordez des autorisations de gestion d'alias uniquement aux principaux qui en ont besoin et limitez leurs autorisations de gestion d'alias aux alias qu'ils doivent gérer.

Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS

Vous pouvez autoriser des utilisateurs ou des rôles associés à un autre Compte AWS à utiliser une clé KMS de votre compte. L'accès inter-comptes nécessite une autorisation dans la politique de clé de la clé KMS et dans une politique IAM dans le compte de l'utilisateur externe.

L'autorisation inter-comptes n'est effective que pour les opérations suivantes :

- Opérations cryptographiques
- CreateGrant
- DescribeKey
- GetKeyRotationStatus
- GetPublicKey
- ListGrants
- RetireGrant
- RevokeGrant

Si vous accordez à un utilisateur d'un autre compte des autorisations pour d'autres opérations, ces autorisations n'ont aucun effet. Par exemple, si vous accordez au principal d'un autre compte une ListKeys autorisation kms : dans une politique IAM, ou kms : une ScheduleKeyDeletion autorisation sur une clé KMS dans une politique clé, les tentatives de l'utilisateur pour appeler ces opérations sur vos ressources échouent toujours.

Pour plus de détails sur l'utilisation des clés KMS dans différents comptes pour les opérations AWS KMS, veuillez consulter la colonne Cross-account use (Utilisation inter-comptes) dans le AWS KMS autorisations et Utilisation de clés KMS dans d'autres comptes. Il existe aussi une section Cross-account use (Utilisation inter-comptes) dans chaque description d'API de la référence d'API AWS Key Management Service.

> **Warning**
>
> Soyez prudent lorsque vous autorisez les principaux à utiliser vos clés KMS. Dans la mesure du possible, suivez le principe du moindre privilège. Donnez uniquement aux utilisateurs l'accès aux clés KMS dont ils ont besoin pour les opérations dont ils ont besoin.
> Par ailleurs, soyez prudent en ce qui concerne l'utilisation d'une clé KMS inconnue, en particulier d'une clé KMS dans un compte différent. Les utilisateurs malveillants peuvent vous autoriser à utiliser leur clé KMS pour obtenir des informations sur vous ou votre compte.
> Pour plus d'informations sur l'utilisation des politiques pour protéger les ressources de votre compte, veuillez consulter Bonnes pratiques pour les politiques IAM.

Pour accorder l'autorisation d'utiliser une clé KMS aux utilisateurs et aux rôles d'un autre compte, vous devez utiliser deux types de politiques différents :

- La politique de clé pour la clé KMS doit accorder au compte externe (ou aux utilisateurs et rôles du compte externe) l'autorisation d'utiliser la clé KMS. La politique de clé se trouve dans le compte qui possède la clé KMS.
- Les politiques IAM du compte externe doivent déléguer les autorisations de politique de clé à leurs utilisateurs et rôles. Ces politiques sont définies dans le compte externe et accordent des autorisations aux utilisateurs et rôles de ce compte.

La politique de clé détermine qui peut avoir accès à la clé KMS. La politique IAM détermine qui a accès à la clé KMS. Ni la politique de clé ni la politique IAM à elles seules ne suffisent. Vous devez modifier les deux.

Pour modifier la politique clé, vous pouvez utiliser la vue des politiques dans les opérations AWS Management Console ou utiliser les PutKeyPolicyopérations CreateKeyou. Pour obtenir de l'aide concernant la définition de la politique de clé lors de la création d'une clé KMS, veuillez consulter Création de clés KMS que d'autres comptes peuvent utiliser.

Pour obtenir de l'aide concernant la modification des politiques IAM, veuillez consulter Utilisation des politiques IAM avec AWS KMS.

Pour obtenir un exemple qui montre comment la politique de clé et les politiques IAM fonctionnent ensemble pour autoriser l'utilisation d'une clé KMS dans un autre compte, veuillez consulter Exemple 2 : l'utilisateur endosse un rôle avec l'autorisation d'utiliser une clé KMS dans un autre Compte AWS.

Vous pouvez afficher les opérations AWS KMS inter-comptes résultantes sur la clé KMS dans vos journaux AWS CloudTrail. Les opérations qui utilisent des clés KMS dans d'autres comptes sont journalisées à la fois dans le compte de l'appelant et dans le compte propriétaire de la clé KMS.

Rubriques

- Étape 1 : ajouter une déclaration de politique de clé dans le compte local
- Étape 2 : ajouter des politiques IAM dans le compte externe
- Création de clés KMS que d'autres comptes peuvent utiliser
- Autoriser l'utilisation de clés KMS externes avec Services AWS
- Utilisation de clés KMS dans d'autres comptes

> **Note**
>
> Les exemples de cette rubrique montrent comment utiliser ensemble une politique de clé et une politique IAM pour fournir et limiter l'accès à une clé KMS. Ces exemples génériques ne sont pas destinés à représenter les autorisations que n'importe quel Service AWS particulier exige sur une clé KMS. Pour de plus amples informations sur les autorisations qu'exige un Service AWS, veuillez consulter la rubrique de chiffrement dans la documentation du service.

Étape 1 : ajouter une déclaration de politique de clé dans le compte local

La politique de clé pour une clé KMS constitue l'élément principal qui détermine qui peut accéder à la clé KMS et quelles sont les opérations pouvant être effectuées. La politique de clé est toujours définie dans le compte propriétaire de la clé KMS. Contrairement aux politiques IAM, les politiques de

clé ne spécifient pas de ressource. La ressource est la clé KMS associée à la politique de clé. Lors de l'octroi d'une autorisation entre comptes, la politique de clé relative à la clé KMS doit accorder au compte externe (ou aux utilisateurs et rôles du compte externe) l'autorisation d'utiliser la clé KMS.

Pour accorder à un compte externe l'autorisation d'utiliser la clé KMS, ajoutez une instruction à la politique de clé qui spécifie le compte externe. Dans l'élément `Principal` de la politique de clé, entrez l'Amazon Resource Name (ARN) du compte externe.

Lorsque vous spécifiez un compte externe dans une politique de clé, les administrateurs IAM du compte externe peuvent utiliser des politiques IAM pour déléguer ces autorisations à tous les utilisateurs et rôles du compte externe. Ils peuvent également décider quelles sont les actions spécifiées dans la politique de clé que les utilisateurs et les rôles peuvent effectuer.

Les autorisations accordées au compte externe et à ses principaux ne sont efficaces que si le compte externe est activé dans la région qui héberge la clé KMS et sa politique de clé. Pour plus d'informations sur les régions qui ne sont pas activées par défaut (« Régions d'adhésion »), veuillez consulter Gestion de Régions AWS dans la Références générales AWS.

Par exemple, supposons que vous vouliez autoriser le compte `444455556666` à utiliser une clé KMS de chiffrement symétrique dans le compte `111122223333`. Pour ce faire, ajoutez une instruction de politique comme celle de l'exemple suivant à la politique de clé pour la clé KMS dans le compte `111122223333`. Cette instruction de politique accorde au compte externe, `444455556666`, l'autorisation d'utiliser la clé KMS dans les opérations de chiffrement pour les clés KMS de chiffrement symétriques.

> **Note**
>
> L'exemple suivant illustre une politique de clé qui permet de partager une clé KMS avec un autre compte. Remplacez les valeurs `Sid`, `Principal` et `Action` de l'exemple par des valeurs valides pour l'utilisation prévue de votre clé KMS.

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::444455556666:root"
        ]
```

```
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Au lieu d'accorder l'autorisation au compte externe, vous pouvez spécifier des utilisateurs et des rôles externes spécifiques dans la politique de clé. Toutefois, ces utilisateurs et rôles ne peuvent pas utiliser la clé KMS tant que les administrateurs IAM du compte externe n'ont pas attaché les politiques IAM appropriées à leurs identités. Les politiques IAM peuvent accorder une autorisation à tous les utilisateurs et rôles externes, ou à une partie d'entre eux seulement, qui sont spécifiés dans la politique de clé. Elles peuvent également autoriser tout ou partie des actions spécifiées dans la politique de clé.

La spécification d'identités dans une politique de clé restreint les autorisations que les administrateurs IAM du compte externe peuvent fournir. Toutefois, cela rend la gestion des politiques avec deux comptes plus complexe. Par exemple, supposons que vous ayez besoin d'ajouter un utilisateur ou un rôle. Vous devez ajouter cette identité à la politique de clé dans le compte propriétaire de la clé KMS et créer des politiques IAM dans le compte de l'identité.

Pour spécifier des utilisateurs ou des rôles externes spécifiques dans une politique de clé, dans l'élément `Principal`, entrez l'Amazon Resource Name (ARN) d'un utilisateur ou d'un rôle dans le compte externe.

Ainsi, l'exemple d'instruction de politique de clé suivant autorise le rôle `ExampleRole` du compte `444455556666` à utiliser une clé KMS du compte `111122223333`. Cette instruction de politique de clé accorde au compte externe, `444455556666`, l'autorisation d'utiliser la clé KMS dans les opérations de chiffrement pour les clés KMS de chiffrement symétriques.

> **Note**
>
> L'exemple suivant illustre une politique de clé qui permet de partager une clé KMS avec un autre compte. Remplacez les valeurs `Sid`, `Principal` et `Action` de l'exemple par des valeurs valides pour l'utilisation prévue de votre clé KMS.

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

> **Note**
>
> Ne définissez pas le principal sur un astérisque (*) dans une instruction de politique de clé qui autorise des autorisations, sauf si vous utilisez des conditions pour limiter la stratégie de clé. Un astérisque autorise chaque identité dans chaque Compte AWS à utiliser la clé KMS, sauf si une autre instruction de politique la refuse explicitement. Les utilisateurs dans d'autres Comptes AWS peuvent utiliser votre clé KMS dès qu'ils ont les autorisations correspondantes dans leur propre compte.

Vous devez également décider quelles autorisations vous souhaitez accorder au compte externe. Pour obtenir la liste des autorisations sur les clés KMS, veuillez consulter AWS KMS autorisations.

Vous pouvez accorder au compte externe l'autorisation d'utiliser la clé KMS dans les opérations cryptographiques et avec les services AWS intégrés à AWS KMS. Pour ce faire, utilisez la section Key Users (Utilisateurs de clés) de l'AWS Management Console. Pour plus de détails, veuillez consulter Création de clés KMS que d'autres comptes peuvent utiliser.

Pour spécifier d'autres autorisations dans les politiques de clé, modifiez le document de politique de clé. Par exemple, vous pouvez accorder aux utilisateurs l'autorisation de déchiffrer, mais pas de chiffrer, ou l'autorisation d'afficher la clé KMS sans l'utiliser. Pour modifier le document de politique clé, vous pouvez utiliser la vue des politiques dans les AWS Management Console PutKeyPolicyopérations CreateKeyou.

Étape 2 : ajouter des politiques IAM dans le compte externe

La politique de clé du compte propriétaire de la clé KMS définit la plage valide pour les autorisations. Cependant, les utilisateurs et les rôles du compte externe ne peuvent pas utiliser la clé KMS tant que vous n'avez pas attaché des politiques IAM qui délèguent ces autorisations ou utilisé des octrois pour gérer l'accès à la clé KMS. Les politiques IAM sont définies dans le compte externe.

Si la politique de clé accorde l'autorisation au compte externe, vous pouvez attacher des politiques IAM à n'importe quel utilisateur ou rôle du compte. Toutefois, si la politique de clé accorde l'autorisation à des utilisateurs ou des rôles spécifiés, la politique IAM peut uniquement accorder ces autorisations à tous les utilisateurs et rôles spécifiés ou à un sous-ensemble. Si une politique IAM accorde l'accès à la clé KMS à d'autres utilisateurs ou rôles externes, cela n'a aucun effet.

La politique de clé limite également les actions dans la politique IAM. La politique IAM peut déléguer tout ou une partie des actions spécifiées dans la politique de clé. Si la politique IAM répertorie les actions qui ne sont pas spécifiées dans la politique de clé, ces autorisations ne sont pas effectives.

L'exemple de politique IAM suivant autorise le principal à utiliser la clé KMS dans le compte `111122223333` pour les opérations de chiffrement. Pour accorder cette autorisation aux utilisateurs et rôles du compte `444455556666`, attachez la politique aux utilisateurs ou rôles du compte `444455556666`.

> **Note**
>
> L'exemple suivant illustre une politique IAM qui permet de partager une clé KMS avec un autre compte. Remplacez les valeurs `Sid`, `Resource` et `Action` de l'exemple par des valeurs valides pour l'utilisation prévue de votre clé KMS.

```
{
    "Sid": "AllowUseOfKeyInAccount111122223333",
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
```

```
    "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Notez les informations suivantes sur cette politique :

- Contrairement aux politiques de clé, les instructions de politique IAM ne contiennent pas l'élément `Principal`. Dans les politiques IAM, le principal est l'identité à laquelle la politique est attachée.
- L'élément `Resource` de la politique IAM identifie la clé KMS que le principal peut utiliser. Pour spécifier une clé KMS, ajoutez son ARN de clé à l'élément `Resource`.
- Vous pouvez spécifier plusieurs clés KMS dans l'élément `Resource`. Si vous ne spécifiez pas de clés KMS particulières dans l'élément `Resource`, vous pouvez accorder par inadvertance l'accès à plus de clés KMS que prévu.
- Pour autoriser l'utilisateur externe à utiliser la clé KMS avec des services AWS qui s'intègrent à AWS KMS, vous pouvez avoir besoin d'ajouter des autorisations à la politique de clé ou à la politique IAM. Pour plus de détails, veuillez consulter Autoriser l'utilisation de clés KMS externes avec Services AWS.

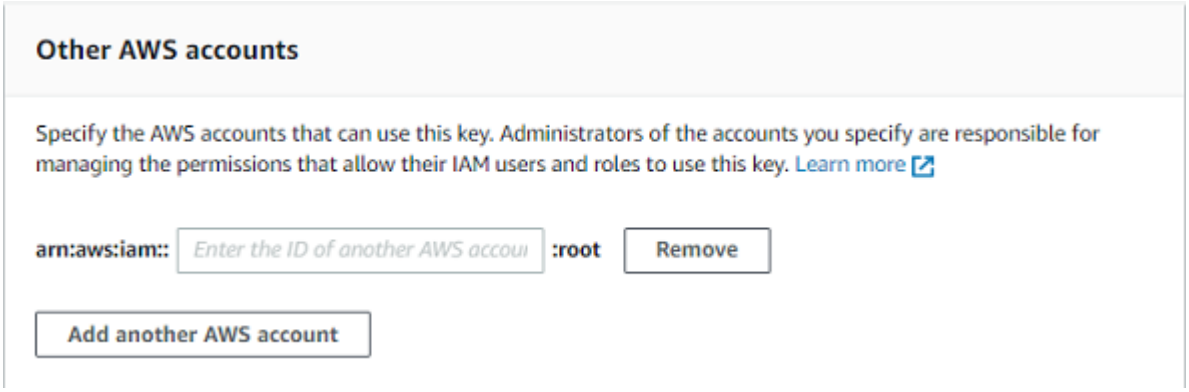
Pour plus d'informations sur l'utilisation des politiques IAM, veuillez consulter Politiques IAM.

Création de clés KMS que d'autres comptes peuvent utiliser

Lorsque vous utilisez l'CreateKeyopération pour créer une clé KMS, vous pouvez utiliser son `Policy` paramètre pour spécifier une politique de clé qui autorise un compte externe, ou des utilisateurs et des rôles externes, à utiliser la clé KMS. Vous devez également ajouter des politiques IAM dans le compte externe qui délèguent ces autorisations aux utilisateurs et rôles du compte, même lorsque des utilisateurs et des rôles sont spécifiés dans la politique de clé. Vous pouvez modifier la politique clé à tout moment en utilisant cette PutKeyPolicyopération.

Lorsque vous créez une clé KMS dans la AWS Management Console, vous créez également sa politique de clé. Lorsque vous sélectionnez des identités dans les sections Key Administrators (Administrateurs de clé) et Key Users (Utilisateurs de clé), AWS KMS ajoute des instructions de politique pour ces identités à la politique de clé de la clé KMS.

La section Key Users (Utilisateurs de clé) vous permet également d'ajouter des comptes externes en tant qu'utilisateurs de clé.



Lorsque vous entrez l'ID de compte d'un compte externe, AWS KMS ajoute deux instructions à la politique de clé. Cette action affecte uniquement la politique de clé. Les utilisateurs et les rôles du compte externe ne peuvent pas utiliser la clé KMS tant que vous n'avez pas attaché de politiques IAM pour leur accorder tout ou une partie de ces autorisations.

La première instruction de politique de clé accorde au compte externe l'autorisation d'utiliser la clé KMS dans les opérations de chiffrement.

> **Note**
>
> Les exemples suivants illustrent une politique de clé qui permet de partager une clé KMS avec un autre compte. Remplacez les valeurs `Sid`, `Principal` et `Action` de l'exemple par des valeurs valides pour l'utilisation prévue de votre clé KMS.

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

La deuxième instruction de politique de clé autorise le compte externe à créer, afficher et révoquer des octrois sur la clé KMS, mais uniquement lorsque la demande provient d'un service AWS intégré à AWS KMS. Ces autorisations permettent à d'autres services AWS qui chiffrent les données utilisateur d'utiliser la clé KMS.

Ces autorisations sont conçues pour les clés KMS qui chiffrent les données utilisateur dans AWS des services tels qu'Amazon WorkMail. Ces services utilisent généralement des octrois pour obtenir les autorisations dont ils ont besoin pour utiliser la clé KMS au nom de l'utilisateur. Pour plus de détails, veuillez consulter Autoriser l'utilisation de clés KMS externes avec Services AWS.

```
{
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": "true"
        }
    }
}
```

Si ces autorisations ne répondent pas à vos besoins, vous pouvez les modifier dans la vue des politiques de la console ou en utilisant l'PutKeyPolicyopération. Vous pouvez spécifier des utilisateurs et des rôles externes particuliers au lieu d'accorder l'autorisation au compte externe. Vous pouvez modifier les actions spécifiées par la politique. Vous pouvez également utiliser des conditions globales et de politique AWS KMS pour affiner les autorisations.

Autoriser l'utilisation de clés KMS externes avec Services AWS

Vous pouvez autoriser un utilisateur d'un autre compte à utiliser votre clé KMS avec un service intégré à AWS KMS. Par exemple, un utilisateur d'un compte externe peut utiliser votre clé KMS pour chiffrer les objets dans un compartiment Amazon S3 ou chiffrer les secrets stockés dans AWS Secrets Manager.

La politique de clé doit accorder à l'utilisateur externe ou au compte de l'utilisateur externe l'autorisation d'utiliser la clé KMS. De plus, vous devez attribuer des politiques IAM à l'identité qui autorise l'utilisateur à utiliser Service AWS. Le service peut également exiger que les utilisateurs disposent d'autorisations supplémentaires dans la politique de clé ou la politique IAM. Pour obtenir la liste des autorisations requises par Service AWS sur une clé gérée par le client, consultez la rubrique Protection des données dans le chapitre Sécurité du guide de l'utilisateur ou du guide du développeur du service.

Utilisation de clés KMS dans d'autres comptes

Si vous avez l'autorisation d'utiliser une clé KMS dans un Compte AWS différent, vous pouvez utiliser la clé KMS dans la AWS Management Console, les kits SDK AWS, la AWS CLI et AWS Tools for PowerShell.

Pour identifier une clé KMS dans un compte différent dans une commande shell ou une demande d'API, utilisez les identificateurs de clé suivants.

- Pour les opérations cryptographiques, et DescribeKeyGetPublicKey, utilisez l'ARN de la clé ou l'alias ARN de la clé KMS.
- Pour CreateGrant, GetKeyRotationStatusListGrants, et RevokeGrant, utilisez l'ARN de la clé KMS.

Si vous saisissez uniquement un ID de clé ou un nom d'alias, AWS suppose que la clé KMS se trouve dans votre compte.

La console AWS KMS n'affiche pas les clés KMS dans d'autres comptes, même si vous avez l'autorisation de les utiliser. En outre, les listes de clés KMS affichées dans les consoles d'autres services AWS n'incluent pas de clés KMS dans d'autres comptes.

Pour spécifier une clé KMS dans un compte différent dans la console d'un service AWS, vous devez entrer l'ARN ou l'alias ARN de la clé KMS. L'identificateur de clé requis varie en fonction du service et peut différer entre la console de service et ses opérations d'API. Pour plus de détails, consultez la documentation du service.

Utilisation des rôles liés aux services pour AWS KMS

AWS Key Management Service utilise des rôles AWS Identity and Access Management (IAM) liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à AWS KMS. Les

rôles liés à un service sont définis par AWS KMS et comprennent toutes les autorisations nécessaires au service pour appeler d'autres services AWS en votre nom.

Un rôle lié à un service permet d'utiliser AWS KMS plus facilement, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. AWS KMS définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul AWS KMS peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable des ressources connexes. Vos ressources AWS KMS sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services où Oui figure dans la colonne Rôle lié à un service. Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations des rôles liés à un service pour les magasins de clés personnalisés AWS KMS

AWS KMSutilise un rôle lié à un service nommé AWSServiceRoleForKeyManagementServiceCustomKeyStorespour prendre en charge les magasins de clés personnalisés. Ce rôle lié à un service accorde à AWS KMS l'autorisation d'afficher vos clusters AWS CloudHSM et de créer l'infrastructure réseau pour prendre en charge une connexion entre votre magasin de clés personnalisé et son clusterAWS CloudHSM. AWS KMS crée ce rôle uniquement lorsque vous créez un magasin de clés personnalisé. Vous ne pouvez pas créer directement ce rôle lié à un service.

Le rôle lié à un service AWSServiceRoleForKeyManagementServiceCustomKeyStores approuve `cks.kms.amazonaws.com` pour qu'il endosse le rôle. Par conséquent, uniquement AWS KMS peut endosser ce rôle lié à un service.

Les autorisations du rôle sont limitées aux actions exécutées par AWS KMS pour connecter un magasin de clés personnalisé à un cluster AWS CloudHSM. Aucune autre autorisation n'est accordée à AWS KMS. Par exemple, AWS KMS n'a pas l'autorisation de créer, gérer ou supprimer vos clusters AWS CloudHSM, vos HSM ou vos sauvegardes.

Pour plus d'informations sur le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStores, y compris la liste des autorisations et des instructions pour afficher le rôle, modifier sa description, le

supprimer et le faire recréer automatiquement par AWS KMS, consultez Autoriser AWS KMS à gérer AWS CloudHSM et les ressources Amazon EC2.

Autorisations des rôles liés à un service pour les clés multi-région AWS KMS

AWS KMSutilise un rôle lié à un service nommé AWSServiceRoleForKeyManagementServiceMultiRegionKeyspour prendre en charge les clés multirégionales. Ce rôle lié à un service accorde à AWS KMS l'autorisation de synchroniser les changements des éléments de clé d'une clé principale multi-région avec les clés de réplica. AWS KMS crée ce rôle uniquement lorsque vous créez une clé principale multi-région. Vous ne pouvez pas créer directement ce rôle lié à un service.

Le rôle lié à un service AWSServiceRoleForKeyManagementServiceMultiRegionKeys approuve `mrk.kms.amazonaws.com` pour qu'il endosse le rôle. Par conséquent, uniquement AWS KMS peut endosser ce rôle lié à un service. Les autorisations du rôle sont limitées aux actions exécutées par AWS KMS pour synchroniser les éléments de clé dans les clés multi-région associées. Aucune autre autorisation n'est accordée à AWS KMS.

Pour plus d'informations sur le rôle AWSServiceRoleForKeyManagementServiceMultiRegionKeys, y compris la liste des autorisations et des instructions pour afficher le rôle, modifier sa description, le supprimer et le faire recréer automatiquement par AWS KMS, consultez Autoriser AWS KMS à synchroniser de clés multi-région.

Mises à jour AWS KMS vers des politiques gérées par AWS

Consultez le détail des mises à jour des politiques gérées par AWS pour AWS KMS depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques sur les modifications apportées à cette page, abonnez-vous au flux RSS de la page AWS KMS Historique du document.

Modification	Description	Date
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy – Mise à jour de la politique existante	AWS KMSa ajouté les `ec2:DescribeNetworkInterfaces` autorisations `ec2:Descr`	10 novembre 2023

Modification	Description	Date
	`ibeVpcs ec2:Descr ibeNetworkAcls` , et pour surveiller les modifications apportées au VPC qui contient votre AWS CloudHSM cluster afin de AWS KMS pouvoir fournir des messages d'erreur clairs en cas de défaillance.	
AWS KMS a démarré le suivi des modifications	AWS KMS a commencé à suivre les modifications pour ses politiques gérées par AWS.	10 novembre 2023

Utilisation du TLS post-quantique hybride avec AWS KMS

AWS Key Management Service (AWS KMS) prend en charge une option d'échange de clés post-quantiques hybrides pour le protocole de chiffrement réseau TLS (Transport Layer Security). Vous pouvez utiliser cette option TLS lorsque vous vous connectez aux points de terminaison de l'API AWS KMS. Nous offrons cette fonctionnalité avant la standardisation des algorithmes post-quantiques afin que vous puissiez commencer à tester l'effet de ces protocoles d'échange de clés sur les appels AWS KMS. Ces fonctionnalités optionnelles d'échange de clés post-quantiques hybrides sont au moins aussi sécurisées que le chiffrement TLS que nous utilisons aujourd'hui et sont susceptibles d'offrir des avantages supplémentaires en matière de sécurité à long terme. Cependant, elles affectent la latence et le débit par rapport aux protocoles d'échange de clés classiques utilisés aujourd'hui.

Les données que vous envoyez à AWS Key Management Service (AWS KMS) sont protégées en transit par le chiffrement fourni par une connexion TLS (Transport Layer Security). Les suites de chiffrement classiques que AWS KMS prend en charge pour les sessions TLS rendent les attaques de force brute contre les mécanismes d'échange clés irréalisables avec la technologie actuelle. Cependant, si l'informatique quantique à grande échelle devient courante à l'avenir, les suites de chiffrement classiques utilisées dans les mécanismes d'échange de clés TLS seront sensibles à ces attaques. Si vous développez des applications qui s'appuient sur la confidentialité à long terme des données transmises via une connexion TLS, vous devriez envisager de migrer vers le chiffrement

post-quantique avant que les ordinateurs quantiques à grande échelle ne deviennent disponibles. AWS travaille à préparer cet avenir, et nous souhaitons que vous soyez bien préparé.

Pour protéger les données chiffrées aujourd'hui contre d'éventuelles attaques futures, AWS participe avec la communauté cryptographique au développement d'algorithmes quantiques ou post-quantiques . Nous avons mis en place des suites de chiffrement d'échange de clés post-quantiques hybrides qui AWS KMS combinent des éléments classiques et post-quantiques afin de veiller à ce que votre connexion TLS soit au moins aussi sûre qu'avec les suites de chiffrement classiques.

Ces suites de chiffrement hybrides sont disponibles pour une utilisation sur vos applications de production dans la plupart des Régions AWS. Cependant, étant donné que les caractéristiques de performance et les exigences de bande passante des suites de chiffrement hybrides sont différentes de celles des mécanismes d'échange de clés classiques, nous vous recommandons de les tester sur vos appels d'API AWS KMS dans des conditions différentes.

Commentaires

Comme toujours, nous accueillons vos commentaires et votre participation à nos référentiels open-source. Nous aimerions en particulier savoir comment votre infrastructure interagit avec cette nouvelle variante du trafic TLS.

- Pour nous faire part de vos commentaires sur ce point, utilisez le lien Commentaires dans le coin supérieur droit de cette page.
- Nous développons ces suites de chiffrement hybrides en open source dans le s2n-tlsréférentiel sur GitHub. Pour fournir des commentaires sur la facilité d'utilisation des suites de chiffrement, ou partager de nouvelles conditions ou résultats de test, créez un problème dans le référentiel s2n-tls.
- Nous écrivons des exemples de code pour utiliser le protocole TLS post-quantique hybride AWS KMS dans le aws-kms-pq-tls-example GitHubréférentiel. Pour poser des questions ou partager des idées sur la configuration de votre client HTTP ou du client AWS KMS pour utiliser les suites de chiffrement hybrides, créez un problème dans le référentiel aws-kms-pq-tls-example.

Régions AWS prises en charge

Post-Quantum TLS pour AWS KMS est disponible dans toutes les Régions AWS que AWS KMS prend en charge excepté pour la Chine (Beijing) et Chine (Ningxia).

> **Note**
>
> AWS KMS ne prend pas en charge le protocole TLS post-quantique hybride pour les points de terminaison FIPS dans AWS GovCloud (US).

Pour obtenir la liste de tous les points de terminaison AWS KMS dans chaque Région AWS, consultez les Points de terminaison et quotas AWS Key Management Service dans le Référence générale d'Amazon Web Services. Pour plus d'informations sur les points de terminaison FIPS, consultez Points de terminaison FIPS dans le Référence générale d'Amazon Web Services.

À propos de l'échange de clés post-quantiques hybrides dans TLS

AWS KMS prend en charge les suites hybrides de chiffrement d'échange de clés post-quantiques. Vous pouvez utiliser les systèmes AWS SDK for Java 2.x et AWS Common Runtime sur Linux pour configurer un client HTTP qui utilise ces suites de chiffrement. Ensuite, chaque fois que vous vous connectez à un point de terminaison AWS KMS avec votre client HTTP, les suites de chiffrement hybrides sont utilisées.

Ce client HTTP utilise s2n-tls, qui est une implémentation open source du protocole TLS. Les suites de chiffrement hybrides que s2n-tls utilise ne sont implémentées qu'à des fins d'échange de clés, et non pour le chiffrement direct des données. Pendant l'échange de clés, le client et le serveur calculent la clé qu'ils utiliseront pour chiffrer et déchiffrer les données sur le réseau.

Les algorithmes utilisés par s2n-tls sont hybrides et combinent Diffie-Hellman (ECDH), un algorithme classique d'échange de clés utilisé aujourd'hui dans TLS, avecKyber, algorithme de chiffrement à clé publique et d'établissement de clés que le National Institute for Standards and Technology (NIST) a désigné comme première norme algorithme post-quantique d'accord de clés. Ce mécanisme hybride utilise chacun des algorithmes indépendamment pour générer une clé. Ensuite, il combine les deux clés cryptographiquement. Avec s2n-tls, vous pouvez configurer un client HTTP de manière à ce qu'il privilégie le protocole TLS post-quantique, ce qui place ECDH avec Kyber en tête de la liste des préférences. Les algorithmes d'échange de clés classiques sont inclus dans la liste des préférences pour garantir la compatibilité, mais ils sont plus bas dans l'ordre de préférence.

Si les recherches en cours révèlent que l'algorithme Kyber n'a pas la force post-quantique attendue, la clé hybride est toujours au moins aussi forte que la seule clé ECDH actuellement utilisée. Jusqu'à ce que ce processus soit terminé, nous recommandons d'utiliser des algorithmes hybrides plutôt que des algorithmes post-quantiques seuls.

Utilisation du TLS post-quantique hybride avec AWS KMS

Vous pouvez utiliser le TLS post-quantique hybride pour vos appels vers AWS KMS. Lors de la configuration de votre environnement de test client HTTP, tenez compte des informations suivantes :

Chiffrement en transit

Les suites de chiffrement hybrides dans s2n-tls sont utilisées uniquement pour le chiffrement en transit. Elles protègent vos données pendant qu'elles se déplacent de votre client vers le point de terminaison AWS KMS. AWS KMS n'utilise pas ces suites de chiffrement pour chiffrer les données sous AWS KMS keys.

Au lieu de cela, lorsque AWS KMS chiffre vos données sous des clés KMS, il utilise un chiffrement symétrique avec des clés 256 bits et l'algorithme Advanced Encryption Standard in Galois Counter Mode (AES-GCM), qui est déjà résistant à la quantique. Dans un avenir théorique, les attaques de calcul quantique à grande échelle sur les textes chiffrés créés sous les clés AES-GCM 256 bits réduiront la sécurité effective de la clé à 128 bits. Ce niveau de sécurité est suffisant pour rendre les attaques de force brute sur les textes chiffrés AWS KMS irréalisables.

Systèmes pris en charge

L'utilisation des suites de chiffrement hybrides dans s2n-tls n'est actuellement prise en charge que sur les systèmes Linux. En outre, ces suites de chiffrement sont prises en charge uniquement dans les kits SDK qui prennent en charge le moteur d'exécution common AWS, par exemple AWS SDK for Java 2.x. Pour obtenir un exemple, consultez Comment configurer le TLS post-quantique hybride.

Points de terminaison AWS KMS

Lorsque vous utilisez les suites de chiffrement hybrides, utilisez le point de terminaison AWS KMS standard. Les suites de chiffrement hybrides dans s2n-tls ne sont pas compatibles avec les points de terminaison validés FIPS 140-2 pour AWS KMS.

Lorsque vous configurez un client HTTP de manière à ce qu'il privilégie les connexions TLS post-quantiques avec s2n-tls, les chiffrements post-quantiques apparaissent en premier dans la liste des préférences de chiffrement. Toutefois, la liste des préférences inclut les chiffrements classiques non hybrides plus bas dans l'ordre de préférence pour la compatibilité. Lorsque vous configurez un client HTTP de manière à ce qu'il privilégie le protocole TLS post-quantique avec un point de terminaison validé par AWS KMS FIPS 140-2, s2n-tls négocie un chiffrement d'échange de clés classique et non hybride.

Pour obtenir la liste de tous les points de terminaison AWS KMS dans chaque Région AWS, consultez les Points de terminaison et quotas AWS Key Management Service dans le Référence générale d'Amazon Web Services. Pour plus d'informations sur les points de terminaison FIPS, consultez Points de terminaison FIPS dans le Référence générale d'Amazon Web Services.

Performances attendues

Nos premiers tests de référence montrent que les suites de chiffrement hybrides dans s2n-tls sont plus lentes que les suites de chiffrement TLS classiques. L'effet varie en fonction du profil réseau, de la vitesse du processeur, du nombre de cœurs et de votre fréquence d'appel. Pour obtenir les résultats des tests de performance, reportez-vous sur Comment régler TLS pour la cryptographie post-quantique hybride avec Kyber.

Comment configurer le TLS post-quantique hybride

Dans cette procédure, ajoutez une dépendance Maven pour le client HTTP AWS Common Runtime. Vous pouvez ensuite configurer un client HTTP qui privilégie le protocole TLS post-quantique. Ensuite, créez un client AWS KMS qui utilise le client HTTP.

Pour voir des exemples complets de configuration et d'utilisation de TLS post-quantique hybride avec AWS KMS, veuillez consulter le référentiel aws-kms-pq-tls-example.

> **Note**
>
> Le client HTTP AWS Common Runtime, qui était disponible en version préliminaire, est désormais généralement disponible depuis février 2023. Dans cette version, la classe `tlsCipherPreference` et le paramètre de méthode `tlsCipherPreference()` sont remplacés par le paramètre de méthode `postQuantumTlsEnabled()`. Si vous utilisiez cet exemple dans la version préliminaire, vous devez procéder à la mise à jour de votre code.

1. Ajoutez le client de moteur d'exécution commun AWS à vos dépendances Maven. Nous vous recommandons d'utiliser la dernière version disponible.

   Par exemple, cette instruction ajoute la version `2.20.0` du client AWS Common Runtime à vos dépendances Maven.

   ```
   <dependency>
       <groupId>software.amazon.awssdk</groupId>
   ```

```
    <artifactId>aws-crt-client</artifactId>
    <version>2.20.0</version>
</dependency>
```

2. Pour activer les suites de chiffrement post-quantique hybrides, ajoutez le AWS SDK for Java 2.x à votre projet et initialisez-le. Activez ensuite les suites de chiffrement post-quantique hybrides sur votre client HTTP, comme indiqué dans l'exemple suivant.

   Ce code utilise le paramètre de méthode `postQuantumTlsEnabled()` pour configurer un client HTTP AWS Common Runtime qui privilégie la suite de chiffrement post-quantique hybride recommandée, ECDH avec Kyber. Il utilise ensuite le client HTTP configuré pour créer une instance du client asynchrone AWS KMS, `KmsAsyncClient`. Une fois ce code terminé, toutes les demandes d'API AWS KMS effectuées sur l'instance `KmsAsyncClient` utilisent le protocole TLS post-quantique hybride.

   ```
   // Configure HTTP client
   SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
              .postQuantumTlsEnabled(true)
              .build();

   // Create the AWS KMS async client
   KmsAsyncClient kmsAsync = KmsAsyncClient.builder()
             .httpClient(awsCrtHttpClient)
             .build();
   ```

3. Testez vos appels AWS KMS avec le protocole TLS post-quantique hybride.

   Lorsque vous appelez des opérations d'API AWS KMS sur le client AWS KMS configuré, vos appels sont transmis au point de terminaison AWS KMS à l'aide de TLS post-quantique hybride. Pour tester votre configuration, appelez une API AWS KMS, telle que `ListKeys`.

   ```
   ListKeysReponse keys = kmsAsync.listKeys().get();
   ```

Tests de TLS post-quantiques hybrides avec AWS KMS

Envisagez d'exécuter les tests suivants avec des suites de chiffrement hybrides sur vos applications qui appellent AWS KMS.

- Exécutez des tests de charge et des repères. Les suites de chiffrement hybrides fonctionnent différemment des algorithmes d'échange de clés traditionnels. Il se peut que vous deviez ajuster

les délais d'expiration de votre connexion pour tenir compte des durées de liaison plus longues. Si vous exécutez à l'intérieur d'une fonction AWS Lambda, étendez le paramètre de délai d'exécution.

- Essayez de vous connecter à partir de différents endroits. Selon le chemin réseau emprunté par votre demande, vous découvrirez peut-être que des hôtes intermédiaires, des proxys ou des pare-feu avec inspection approfondie des paquets (DPI) bloquent la demande. Cela peut être dû à l'utilisation des nouvelles suites de chiffrement dans le ClientHellocadre de la poignée de main TLS ou à des messages d'échange de clés plus volumineux. Si vous rencontrez des difficultés pour résoudre ces problèmes, collaborez avec votre équipe de sécurité ou les administrateurs informatiques pour mettre à jour la configuration appropriée et débloquer les nouvelles suites de chiffrement TLS.

En savoir plus sur le TLS post-quantique dans AWS KMS

Pour de plus amples informations sur l'utilisation de TLS post-quantique hybride dans AWS KMS, veuillez consulter les ressources suivantes.

- Pour en savoir plus sur le chiffrement post-quantique sur AWS, et obtenir des liens vers des articles de blog et des documents de recherche, consultez Chiffrement post-quantique.
- Pour de plus amples informations sur s2n-tls, veuillez consulter Introducing s2n-tls, a New Open Source TLS Implementation et Using s2n-tls.
- Pour plus d'informations sur le client HTTP AWS Common Runtime, consultez Configuring the AWS CRT-based HTTP client (Configuration du client HTTP basé sur CRT) dans le Guide du développeur AWS SDK for Java 2.x.
- Pour de plus amples informations sur le projet de chiffrement post-quantique du National Institute for Standards and Technology (NIST), veuillez consulter Chiffrement post-quantique.
- Pour plus d'informations sur la normalisation du chiffrement post-quantique par le NIST, consultez la page Post-Quantum Cryptography Standardization (Normalisation du chiffrement post-quantique).

Déterminer l'accès à des AWS KMS keys

Pour déterminer globalement qui a actuellement accès à une AWS KMS key, vous devez examiner la politique de la clé KMS, tous les octrois qui s'appliquent à la clé KMS et éventuellement toutes les politiques AWS Identity and Access Management (IAM). Vous pouvez le faire pour déterminer la portée de l'utilisation potentielle d'une clé KMS ou pour mieux répondre aux exigences d'audit ou de

conformité. Les rubriques suivantes peuvent vous aider à générer une liste complète des principaux AWS (identités) qui ont actuellement accès à une clé KMS.

Rubriques

- Examen de la politique de clé
- Examen des politiques IAM
- Examen des octrois
- Résolution des problèmes de clé d'accès

Examen de la politique de clé

Les politiques de clé constituent le principal moyen de contrôler l'accès aux clés KMS. Chaque clé KMS a exactement une politique de clé.

Lorsqu'une politique de clé inclut la politique de clé par défaut, elle permet aux administrateurs IAM du compte d'utiliser des politiques IAM pour contrôler l'accès à la clé KMS. En outre, si la politique de clé donne à un autre Compte AWS l'autorisation d'utiliser la clé KMS, les administrateurs IAM du compte externe peuvent utiliser des politiques IAM pour déléguer ces autorisations. Pour déterminer la liste complète des principaux qui peuvent accéder à la clé KMS, examinez les politiques IAM.

Pour consulter la politique clé d'une clé gérée par le AWS KMS client ou Clé gérée par AWSde votre compte, utilisez l'GetKeyPolicyopération AWS Management Console ou dans l'AWS KMSAPI. Pour afficher la politique de clé, vous devez disposer des autorisations `kms:GetKeyPolicy` pour la clé KMS. Pour obtenir des instructions sur l'affichage de la politique de clé d'une clé KMS, veuillez consulter the section called “Affichage d'une politique de clé”.

Examinez le document de politique de clé et notez tous les principaux spécifiés dans l'élément `Principal` de chaque instruction de politique. Dans une instruction de politique disposant d'un effet `Allow`, les utilisateur IAM, les rôles IAM et les Comptes AWS associés à l'élément `Principal` ont accès à cette clé KMS.

> **Note**
>
> Ne définissez pas le principal sur un astérisque (*) dans une instruction de politique de clé qui autorise des autorisations, sauf si vous utilisez des conditions pour limiter la politique de clé. Un astérisque autorise chaque identité dans chaque Compte AWS à utiliser la clé KMS, sauf si une autre instruction de politique la refuse explicitement. Les utilisateurs dans d'autres

Comptes AWS peuvent utiliser votre clé KMS dès qu'ils ont les autorisations correspondantes dans leur propre compte.

Les exemples suivants utilisent les instructions de politique trouvées dans la politique de clé par défaut pour montrer comment procéder.

Example Instruction de politique 1

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
  "Action": "kms:*",
  "Resource": "*"
}
```

Dans l'instruction de politique 1, `arn:aws:iam::111122223333:root` est un principal de compte AWS qui fait référence au Compte AWS 111122223333. (Il ne s'agit pas de l'utilisateur root du compte). Par défaut, une instruction de politique comme celle-ci est incluse dans le document de politique de clé lorsque vous créez une nouvelle clé CMK à l'aide de la AWS Management Console, ou que vous créez une nouvelle clé CMK par programmation sans fournir de politique de clé.

Un document de politique de clé contenant une instruction qui autorise l'accès au Compte AWS permet aux politiques IAM figurant dans le compte d'autoriser l'accès à la clé KMS. Cela signifie que les utilisateurs et rôles figurant dans le compte peuvent avoir accès à la clé KMS, même s'ils ne sont pas répertoriés explicitement en tant que principaux dans le document de politique de clé. Prenez soin d'examiner toutes les politiques IAM dans tous les Comptes AWS répertoriés en tant que principaux pour déterminer si elles autorisent l'accès à cette clé KMS.

Example Instruction de politique 2

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/KMSKeyAdmins"},
  "Action": [
    "kms:Describe*",
    "kms:Put*",
    "kms:Create*",
```

```
    "kms:Update*",
    "kms:Enable*",
    "kms:Revoke*",
    "kms:List*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

Dans la déclaration de politique 2, `arn:aws:iam::111122223333:role/KMSKeyAdmins` fait référence au rôle IAM nommé KMS KeyAdmins dans le document Compte AWS 111122223333. Les utilisateurs autorisés à assumer ce rôle sont habilités à effectuer les opérations répertoriées dans l'instruction de politique, à savoir les opérations administratives permettant de gérer une clé KMS.

Example Instruction de politique 3

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey*",
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Dans la déclaration de politique 3, `arn:aws:iam::111122223333:role/EncryptionApp` fait référence au rôle IAM nommé EncryptionApp dans Compte AWS 111122223333. Les principaux autorisés à assumer ce rôle sont habilités à effectuer les opérations répertoriées dans l'instruction de politique, y compris les opérations de chiffrement relatives à une clé KMS de chiffrement symétrique.

Example Instruction de politique 4

```
{
```

```
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:ListGrants",
    "kms:CreateGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

Dans la déclaration de politique 4, `arn:aws:iam::111122223333:role/EncryptionApp` fait référence au rôle IAM nommé EncryptionApp dans Compte AWS 111122223333. Les principaux autorisés à assumer ce rôle sont habilités à effectuer les opérations répertoriées dans l'instruction de politique. Ces actions, lorsqu'elles sont combinées aux actions autorisées dans l'exemple d'instruction de politique 3, sont celles requises pour déléguer l'utilisation de la clé KMS à la plupart des services AWS qui s'intègrent à AWS KMS, notamment aux services qui utilisent des octrois. La GrantIsFor AWSResource valeur kms : dans l'`Condition`élément garantit que la délégation n'est autorisée que lorsque le délégué est un AWS service qui intègre AWS KMS et utilise des autorisations d'autorisation.

Pour découvrir toutes les différentes façons de spécifier un principal dans un document de politique de clé, veuillez consulter la page Spécification d'un principal dans le Guide de l'utilisateur IAM.

Pour en savoir plus sur les politiques de clé AWS KMS, consultez Politiques de clé dans AWS KMS.

Examen des politiques IAM

Outre la politique de clé et les octrois, vous pouvez utiliser des politiques IAM pour autoriser l'accès à une clé KMS. Pour plus d'informations sur la manière dont les politiques de clé et les politiques IAM fonctionnent ensemble, veuillez consulter Résolution des problèmes de clé d'accès.

Pour déterminer quels principaux ont actuellement accès à une clé KMS via les politiques IAM, vous pouvez utiliser l'outil simulateur de politiques IAM basé sur le navigateur ou vous pouvez adresser des demandes à l'API IAM.

Manières d'examiner les politiques IAM

- Examen des politiques IAM avec le simulateur de politiques IAM
- Examen des politiques IAM avec l'API IAM

Examen des politiques IAM avec le simulateur de politiques IAM

Le simulateur de politiques IAM peut vous aider à découvrir quels principaux ont accès à une clé KMS via une politique IAM.

Pour utiliser le simulateur de politiques IAM pour déterminer l'accès à une clé KMS

1. Connectez-vous à AWS Management Console puis ouvrez le simulateur de politiques IAM à l'adresse https://policysim.aws.amazon.com/.
2. Dans le volet Users, Groups, and Roles, choisissez l'utilisateur, le groupe ou le rôle dont vous souhaitez simuler les politiques.
3. (Facultatif) Décochez les cases en regard de toutes les politiques que vous souhaitez ignorer pour la simulation. Pour simuler toutes les politiques, laissez toutes les politiques sélectionnées.
4. Dans le volet Policy Simulator, procédez comme suit :
   a. Pour Select service, choisissez Key Management Service.
   b. Pour simuler des actions AWS KMS spécifiques, pour Sélectionner des actions, choisissez les actions à simuler. Pour simuler toutes les actions AWS KMS, choisissez Sélectionner tout.
5. (Facultatif) Le simulateur de politiques simule l'accès à toutes les clés KMS par défaut. Pour simuler l'accès à une clé KMS spécifique, sélectionnez Simulation Settings (Paramètres de simulation), puis saisissez l'Amazon Resource Name (ARN) de la clé KMS à simuler.
6. Choisissez Exécuter la simulation.

Vous pouvez afficher les résultats de la simulation dans la section Résultats. Répétez les étapes 2 à 6 pour chaque utilisateur, groupe et rôle figurant dans le Compte AWS.

Examen des politiques IAM avec l'API IAM

Vous pouvez utiliser l'API IAM pour examiner par programmation les politiques IAM. Les étapes suivantes offrent une présentation générale de la façon de procéder :

1. Pour chaque utilisateur Compte AWS répertorié comme principal dans la politique clé (c'est-à-dire chaque principal de AWS compte spécifié dans ce format :`"Principal": {"AWS": "arn:aws:iam::111122223333:root"}`), utilisez les ListRolesopérations ListUserset de l'API IAM pour obtenir tous les utilisateurs et rôles du compte.

2. Pour chaque utilisateur et chaque rôle de la liste, utilisez l'SimulatePrincipalPolicyopération de l'API IAM en transmettant les paramètres suivants :
   - Pour `PolicySourceArn`, spécifiez le nom ARN (Amazon Resource Name) d'un utilisateur ou d'un rôle figurant dans votre liste. Vous ne pouvez spécifier qu'un seul nom `PolicySourceArn` pour chaque demande `SimulatePrincipalPolicy`. Vous devez donc appeler cette opération plusieurs fois, une fois pour chaque utilisateur et rôle de votre liste.
   - Pour la liste `ActionNames`, spécifiez chaque action d'API AWS KMS à simuler. Pour simuler toutes les actions d'API AWS KMS, utilisez `kms:*`. Pour tester les actions d'API AWS KMS individuelles, faites précéder chaque action d'API de « `kms:` », par exemple de « `kms:ListKeys` ». Pour obtenir la liste complète des actions de l'API AWS KMS, consultez Actions dans la référence d'API AWS Key Management Service.
   - (Facultatif) Pour déterminer si les utilisateurs ou les rôles ont accès à des clés KMS spécifiques, utilisez le paramètre `ResourceArns` pour spécifier la liste des Amazon Resource Names (ARN) des clés KMS. Pour déterminer si les utilisateurs ou les rôles ont accès à une clé KMS quelconque, omettez le paramètre `ResourceArns`.

IAM répond à chaque demande `SimulatePrincipalPolicy` avec une décision d'évaluation : `allowed`, `explicitDeny` ou `implicitDeny`. Pour chaque réponse qui contient une décision d'évaluation de `allowed`, la réponse inclut le nom de l'opération d'API AWS KMS spécifique autorisée. Elle inclut également l'ARN de la clé KMS qui a été utilisée dans l'évaluation, le cas échéant.

Examen des octrois

Les octrois sont des mécanismes avancés de spécification d'autorisations que vous ou un service AWS intégré à AWS KMS pouvez utiliser pour spécifier quand et comment une clé KMS peut être utilisée. Les octrois sont attachés à une clé KMS et chaque octroi contient le principal qui reçoit l'autorisation d'utiliser la clé KMS et la liste des opérations autorisées. Les octrois représentent une alternative à la politique de clé et sont utiles pour des cas d'utilisation spécifiques. Pour plus d'informations, consultez Octrois dans AWS KMS.

Pour obtenir une liste des autorisations pour une clé KMS, utilisez l'AWS KMSListGrantsopération. Vous pouvez examiner les octrois définis pour une clé KMS afin de déterminer qui a actuellement accès à la clé KMS pour l'utiliser via ces octrois. Par exemple, ce qui suit est une représentation JSON d'un octroi qui a été obtenu à partir de la commande list-grants dans l'AWS CLI.

```
{"Grants": [{
```

```
  "Operations": ["Decrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "0d8aa621-43ef-4657-b29c-3752c41dc132",
  "RetiringPrincipal": "arn:aws:iam::123456789012:root",
  "GranteePrincipal": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-5d476fab",
  "GrantId": "dc716f53c93acacf291b1540de3e5a232b76256c83b2ecb22cdefa26576a2d3e",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151834E9,
  "Constraints": {"EncryptionContextSubset": {"aws:ebs:id": "vol-5cccfb4e"}}
}]}
```

Pour identifier qui a accès à la clé KMS pour l'utiliser, recherchez l'élément `"GranteePrincipal"`. Dans l'exemple précédent, le principal bénéficiaire est un utilisateur du rôle assumé associé à l'instance EC2 i-5d476fab. L'infrastructure EC2 utilise ce rôle pour attacher le volume EBS chiffré vol-5cccfb4e à l'instance. Dans ce cas, le rôle d'infrastructure EC2 a l'autorisation d'utiliser la clé KMS, parce que vous avez créé précédemment un volume EBS chiffré protégé par cette clé KMS. Puis, vous avez attaché le volume à une instance EC2.

Ce qui suit est un autre exemple de représentation JSON d'un octroi qui a été obtenu à partir de la commande list-grants dans l'AWS CLI. Dans l'exemple suivant, le principal bénéficiaire est un autre Compte AWS.

```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

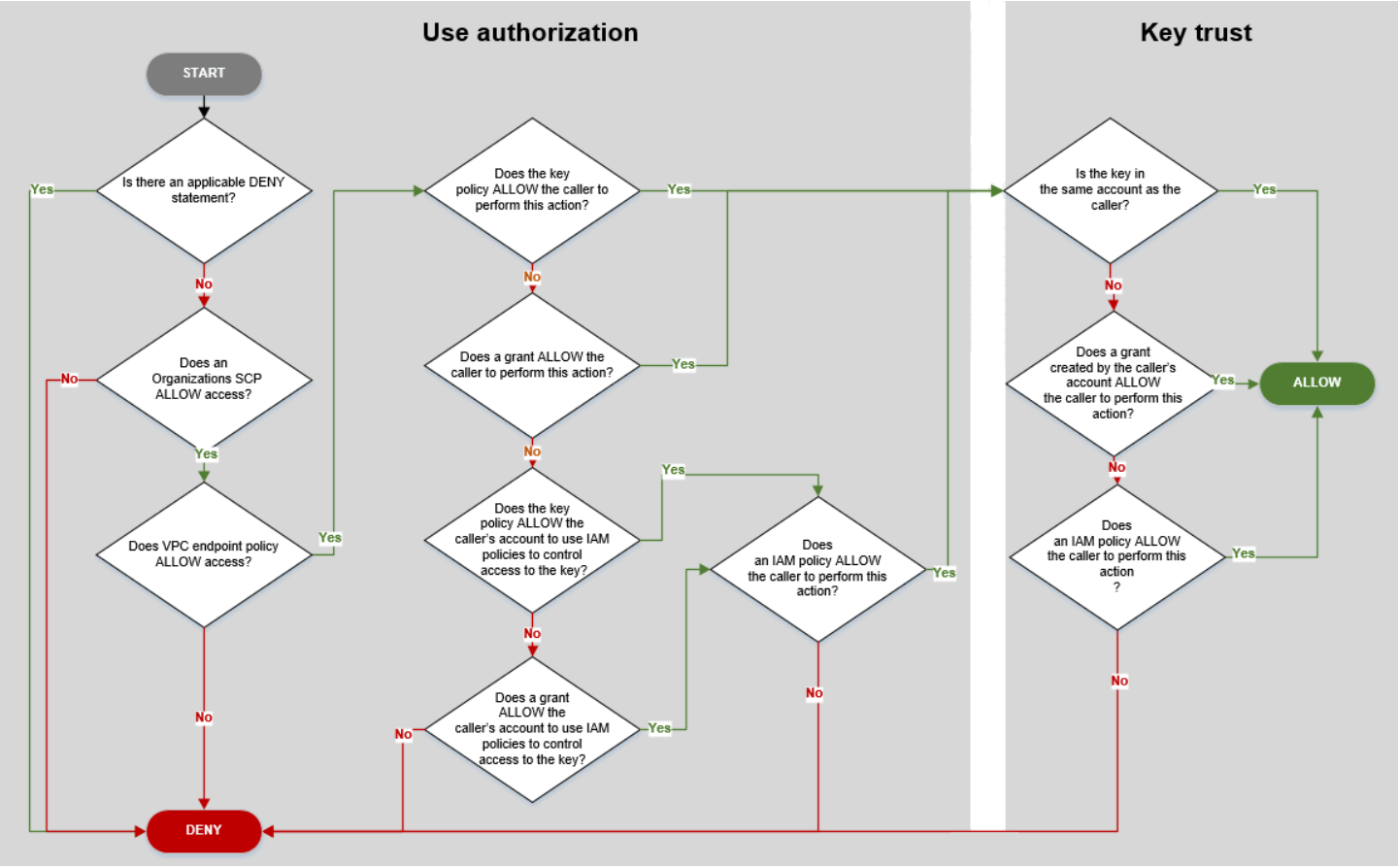
Résolution des problèmes de clé d'accès

Lorsque vous autorisez l'accès à une clé KMS, AWS KMS évalue les éléments suivants :

- La politique de clé attachée à la clé KMS. La politique de clé est toujours définie dans le Compte AWS et la région qui possèdent la clé KMS.

- Toutes les politiques IAM attribuées à l'utilisateur ou au rôle à l'origine de la demande. Les politiques IAM qui régissent l'utilisation d'une clé KMS par un principal sont toujours définies dans le Compte AWS du principal.
- Tous les octrois qui s'appliquent à la clé KMS.
- D'autres types de politiques qui peuvent s'appliquer à la demande d'utilisation de la clé KMS, tels que les politiques de contrôle des services AWS Organizations et les politiques de point de terminaison d'un VPC. Ces politiques sont facultatives et autorisent toutes les actions par défaut, mais vous pouvez les utiliser pour restreindre les autorisations accordées par ailleurs aux principaux.

AWS KMS évalue ces mécanismes de politiques afin de déterminer si l'accès à la clé KMS sera autorisé ou rejeté. Pour ce faire, AWS KMS utilise un processus similaire à celui qui est illustré dans le diagramme suivant. Le diagramme suivant fournit une représentation visuelle du processus d'évaluation de la politique.



Ce diagramme est divisé en deux parties. Les parties semblent être séquentielles, mais elles sont généralement évaluées en même temps.

- Use authorization détermine si vous êtes autorisé à utiliser une clé KMS en fonction de sa politique de clé, des politiques IAM, des octrois et des autres politiques applicables.
- Key trust détermine si vous devez approuver une clé KMS que vous êtes autorisé à utiliser. En général, vous approuvez les ressources de votre Compte AWS. Cependant, vous pouvez également être sûr de l'utilisation de clés KMS dans un autre Compte AWS si un octroi ou une politique IAM de votre compte vous permet d'utiliser la clé KMS.

Vous pouvez utiliser ce diagramme de flux pour découvrir pourquoi un appelant est autorisé ou non à utiliser une clé KMS. Vous pouvez également l'utiliser pour évaluer vos politiques et octrois. Par exemple, le diagramme montre qu'un appelant peut se voir refuser l'accès par une instruction `DENY` explicite, ou en l'absence d'une instruction `ALLOW` explicite, dans la politique de clé, la politique IAM, ou l'octroi.

Le diagramme peut expliquer certains scénarios d'autorisation courants.

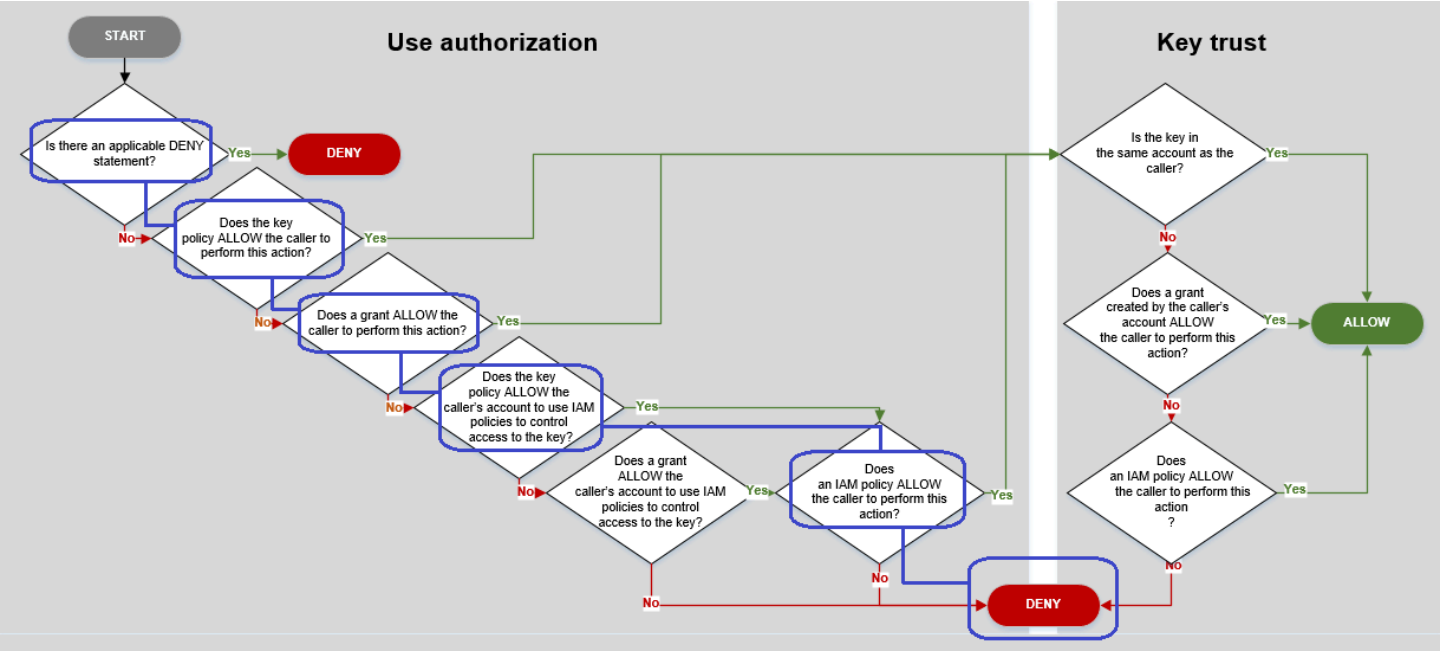
Exemples d'autorisation

- Exemple 1 : l'utilisateur se voit refuser l'accès à une clé KMS sur son Compte AWS
- Exemple 2 : l'utilisateur endosse un rôle avec l'autorisation d'utiliser une clé KMS dans un autre Compte AWS

Exemple 1 : l'utilisateur se voit refuser l'accès à une clé KMS sur son Compte AWS

Alice est une utilisatrice IAM sur le Compte AWS 111122223333. L'accès à une clé KMS lui a été refusé sur ce même Compte AWS. Pourquoi Alice ne peut-elle pas utiliser la clé KMS ?

Dans ce cas, Alice se voit refuser l'accès à la clé KMS, car aucune politique de clé, aucune politique IAM ou aucun octroi ne lui accorde les autorisations requises. La politique de clé de la clé KMS permet au Compte AWS d'utiliser les politiques IAM pour contrôler l'accès à la clé KMS, mais aucune politique IAM n'autorise Alice à utiliser la clé KMS.



Considérez les politiques appropriées dans cet exemple.

- La clé KMS qu'Alice souhaite utiliser dispose de la politique de clé par défaut. Cette politique autorise le Compte AWS qui possède la clé KMS à utiliser des politiques IAM pour contrôler l'accès à la clé KMS. Cette politique de clé remplit la condition La politique de clé AUTORISE-t-elle le compte du principal à utiliser les politiques IAM pour contrôler l'accès à la clé ? du diagramme de flux.

```
{
  "Version" : "2012-10-17",
  "Id" : "key-test-1",
  "Statement" : [ {
    "Sid" : "Delegate to IAM policies",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

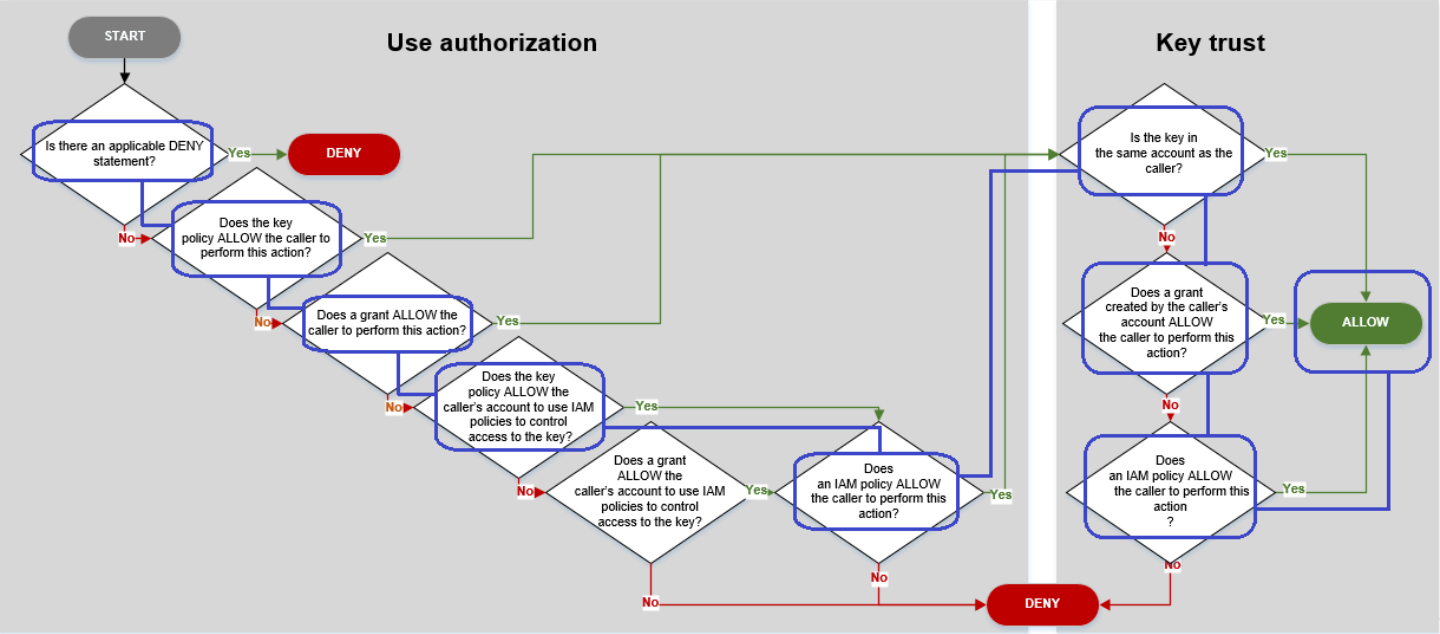
- Cependant, aucune politique de clé, aucune politique IAM ou aucun octroi ne donne à Alice l'autorisation d'utiliser la clé KMS. Par conséquent, Alice se voit refuser l'autorisation d'utiliser la clé KMS.

Exemple 2 : l'utilisateur endosse un rôle avec l'autorisation d'utiliser une clé KMS dans un autre Compte AWS

Bob est un utilisateur du compte 1 (111122223333). Il est autorisé à utiliser une clé KMS du compte 2 (444455556666) pour les opérations cryptographiques. Comment est-ce possible ?

> **Tip**
>
> Lors de l'évaluation des autorisations inter-comptes, n'oubliez pas que la politique de clé est spécifiée dans le compte de la clé KMS. La politique IAM est spécifiée dans le compte de l'appelant, même lorsque l'appelant se trouve dans un autre compte. Pour plus de détails sur la fourniture d'un accès inter-comptes aux clés KMS, veuillez consulter Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

- La politique de clé de la clé KMS du compte 2 autorise ce dernier à utiliser les politiques IAM pour contrôler l'accès à la clé KMS.
- La politique de clé de la clé KMS du compte 2 autorise le compte 1 à utiliser la clé KMS pour les opérations de chiffrement. Toutefois, le compte 1 doit utiliser les politiques IAM pour accorder à ses principaux l'accès à la clé KMS.
- Une politique IAM du compte 1 autorise le rôle `Engineering` à utiliser la clé KMS dans le compte 2 pour les opérations de chiffrement.
- Bob, un utilisateur du compte 1, a l'autorisation d'endosser le rôle `Engineering`.
- Enfin, Bob peut faire confiance à cette clé KMS. En effet, même si cette dernière n'appartient pas à son compte, une politique IAM de son compte lui donne l'autorisation explicite d'utiliser cette clé KMS.



Examinons les politiques qui permettent à Bob, un utilisateur du compte 1, d'utiliser la clé KMS du compte 2.

- La politique de clé de la clé KMS autorise le compte 2 (444455556666, le compte qui possède la clé KMS) à utiliser les politiques IAM pour contrôler l'accès à la clé KMS. Cette politique de clé permet également au compte 1 (111122223333) d'utiliser la clé KMS pour les opérations de chiffrement (spécifiées dans l'élément `Action` de l'instruction de politique). Toutefois, aucun utilisateur du compte 1 ne peut utiliser la clé KMS du compte 2 tant que le compte 1 n'a pas défini les politiques IAM qui accordent aux principaux l'accès à la clé KMS.

  Dans le diagramme de flux, cette politique de clé du compte 2 remplit la condition La politique de clé AUTORISE-t-elle le compte de l'appelant à utiliser les politiques IAM pour contrôler l'accès à la clé ?.

```
{
    "Id": "key-policy-acct-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to use IAM policies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Action": "kms:*",
```

```
            "Resource": "*"
        },
        {
            "Sid": "Allow account 1 to use this KMS key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

- Une politique IAM du compte Compte AWS du principal (compte 1, 111122223333) accorde au principal l'autorisation d'exécuter des opérations de chiffrement à l'aide de la clé KMS dans le compte 2 (444455556666). L'élément `Action` accorde au principal les mêmes autorisations que la politique de clé du compte 2 a accordées au compte 1. Pour accorder ces autorisations au rôle `Engineering` du compte 1, cette politique en ligne est intégrée au rôle `Engineering`.

  De telles politiques IAM inter-comptes sont efficaces uniquement lorsque la politique de clé de la clé KMS du compte 2 accorde au compte 1 l'autorisation d'utiliser la clé KMS. De plus, le compte 1 peut uniquement accorder aux principaux l'autorisation d'effectuer les actions accordées au compte par la politique de clé.

  Dans le diagramme de flux, cela remplit la condition Une politique IAM autorise-t-elle l'appelant à effectuer cette action ?.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:us-
west-2:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            ]
        }
    ]
}
```

- Le dernier élément requis est la définition du rôle `Engineering` dans le compte 1. Le document `AssumeRolePolicyDocument` dans le rôle permet à Bob d'endosser le rôle `Engineering`.

```
{
    "Role": {
        "Arn": "arn:aws:iam::111122223333:role/Engineering",
        "CreateDate": "2019-05-16T00:09:25Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": {
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:user/bob"
                },
                "Effect": "Allow",
                "Action": "sts:AssumeRole"
            }
        },
        "Path": "/",
        "RoleName": "Engineering",
        "RoleId": "AROA4KJY2TU23Y7NK62MV"
    }
}
```

AWS KMS autorisations

Ce tableau est conçu pour vous aider à comprendre AWS KMS les autorisations afin que vous puissiez contrôler l'accès à vos AWS KMS ressources. Les définitions des en-têtes de colonne apparaissent sous le tableau.

Vous pouvez également en savoir plus sur AWS KMS les autorisations dans la AWS Key Management Servicerubrique Actions, ressources et clés de condition de la référence d'autorisation de service. Toutefois, cette rubrique ne répertorie pas toutes les clés de condition que vous pouvez utiliser pour affiner chaque autorisation.

> **Note**
>
> Vous devrez peut-être faire défiler horizontalement ou verticalement pour voir toutes les données de la table.

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
CancelKeyDeletion `kms:CancelKeyDeletion`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
ConnectCustomKeyStore `kms:ConnectCustomKeyStore`	Politique IAM	Non	*	km : CallerAccount

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
CreateAlias `kms:CreateAlias`	Politique IAM (pour l'alias)	Non	Alias	Aucune (lorsque vous contrôlez l'accès à l'alias)
Pour utiliser cette opération, l'appelant doit avoir l'autorisation `kms:CreateAlias` sur deux ressources : • L'alias (dans une politique IAM) • La clé KMS (dans une politique de clé) Pour plus de détails, veuillez consulter Contrôle de l'accès aux alias.	Politique de clé (pour la clé KMS)	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
CreateCustomKeyStore `kms:CreateCustomKeyStore`	Politique IAM	Non	*	km : CallerAccount

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
CreateGrant `kms:CreateGrant`	Stratégie de clé	Oui	Clé KMS	Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions d'octroi : km : GrantConstraintType km : GranteePrincipal km : GrantIsForAWSResource km : GrantOperations km : RetiringPrincipal Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
CreateKey `kms:CreateKey`	Politique IAM	Non	*	km : BypassPolicyLockoutSafetyCheck km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ViaService aws :RequestTag/tag-key (clé de conditionAWS globale) aws :ResourceTag/tag-key (clé de conditionAWS globale) aws : TagKeys (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
Decrypt `kms:Decrypt`	Politique de clé	Oui	Clé KMS	Conditions des opérations de chiffrement km : EncryptionAlgorithm km : RequestAlias Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ViaService
DeleteAlias `kms:DeleteAlias` Pour utiliser cette opération, l'appelant doit avoir l'autorisation `kms:DeleteAlias` sur deux ressources : • L'alias (dans une politique IAM) • La clé KMS (dans une politique de clé) Pour plus de détails, veuillez consulter Contrôle de l'accès aux alias.	Politique IAM (pour l'alias)	Non	Alias	Aucune (lorsque vous contrôlez l'accès à l'alias)
	Politique de clé (pour la clé KMS)	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
DeleteCustomKeyStore `kms:DeleteCustomKeyStore`	Politique IAM	Non	*	km : CallerAccount

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
DeleteImportedKeyMaterial `kms:DeleteImportedKeyMaterial`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
DescribeCustomKeyStores `kms:DescribeCustomKeyStores`	Politique IAM	Non	*	km : CallerAccount

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
DescribeKey `kms:DescribeKey`	Stratégie de clé	Oui	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : RequestAlias

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
DisableKey `kms:DisableKey`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
DisableKeyRotation `kms:DisableKeyRotation`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
DisconnectCustomKeyStore `kms:DisconnectCustomKeyStore`	Politique IAM	Non	*	km : CallerAccount

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
EnableKey `kms:EnableKey`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
EnableKeyRotation `kms:EnableKeyRotation`	Stratégie de clé	Non	Clé KMS (symétrique uniquement)	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
Encrypt `kms:Encrypt`	Politique de clé	Oui	Clé KMS	Conditions des opérations de chiffrement km : EncryptionAlgorithm km : RequestAlias Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GenerateDataKey `kms:GenerateDataKey`	Stratégie de clé	Oui	Clé KMS (symétrique uniquement)	Conditions des opérations de chiffrement km : EncryptionAlgorithm km : RequestAlias Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GenerateDataKeyPair `kms:GenerateDataKeyPair`	Stratégie de clé	Oui	Clé KMS (symétrique uniquement) Génère une paire de clés de données asymétriques qui est protégée par une clé KMS de chiffrement symétrique.	Conditions pour les paires de clés de données : km : DataKeyPairSpec Conditions des opérations de chiffrement km : EncryptionAlgorithm km : RequestAlias Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GenerateDataKeyPairWithoutPlaintext `kms:GenerateDataKeyPairWithoutPlaintext`	Stratégie de clé	Oui	Clé KMS (symétrique uniquement) Génère une paire de clés de données asymétriques qui est protégée par une clé KMS de chiffrement symétrique.	Conditions pour les paires de clés de données : km : DataKeyPairSpec Conditions des opérations de chiffrement km : EncryptionAlgorithm km : RequestAlias Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GenerateDataKeyWithoutPlaintext `kms:GenerateDataKeyWithoutPlaintext`	Stratégie de clé	Oui	Clé KMS (symétrique uniquement)	Conditions des opérations de chiffrement km : EncryptionAlgorithm km : RequestAlias Conditions du contexte de chiffrement : kms EncryptionContext : touche contextuelle km : EncryptionContextKeys Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ViaService
GenerateMac kms:GenerateMac	Stratégie de clé	Oui	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Conditions des opérations de chiffrement : km : MacAlgorithm km : RequestAlias
GenerateRandom kms:GenerateRandom	Politique IAM	N/A	*	Aucun

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GetKeyPolicy `kms:GetKeyPolicy`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GetKeyRotationStatus `kms:GetKeyRotationStatus`	Stratégie de clé	Oui	Clé KMS (symétrique uniquement)	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GetParametersForImport `kms:GetParametersForImport`	Stratégie de clé	Non	Clé KMS	km : WrappingAlgorithm km : WrappingKeySpec Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
GetPublicKey `kms:GetPublicKey`	Stratégie de clé	Oui	Clé KMS (asymétrique uniquement)	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : RequestAlias

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
ImportKeyMaterial `kms:ImportKeyMaterial`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : ExpirationModel km : ValidTo
ListAliases `kms:ListAliases`	Politique IAM	Non	*	Aucun

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
ListGrants `kms:ListGrants`	Stratégie de clé	Oui	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : GrantIsForAWSResource

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
ListKeyPolicies `kms:ListKeyPolicies`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
ListKeys `kms:ListKeys`	Politique IAM	Non	*	Aucun

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
ListResourceTags `kms:ListResourceTags`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
ListRetirableGrants `kms:ListRetirableGrants`	Politique IAM	Le principal spécifié doit être dans le compte local, mais l'opération renvoie des octrois dans tous les comptes.	*	Aucun

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
PutKeyPolicy `kms:PutKeyPolicy`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : BypassPolicyLockoutSafetyCheck

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition

ReEncrypt
 	
`kms:ReEncryptFrom`
 	
`kms:ReEncryptTo`
 	
Pour utiliser cette opération, l'appelant doit avoir l'autorisation sur deux clés KMS :
 	
• `kms:ReEncryptFrom` sur la clé KMS utilisée pour déchiffrer
• `kms:ReEncryptTo` sur la clé KMS utilisée pour chiffrer
	Politique de clé	Oui	Clé KMS	Conditions des opérations de chiffrement
				
				km : EncryptionAlgorithm
				
				km : RequestAlias
				
				Conditions du contexte de chiffrement :
				
				kms EncryptionContext : touche contextuelle
				
				km : EncryptionContextKeys
				
				Conditions pour les opérations de clé KMS :
				
				km : CallerAccount
				
				km : KeySpec
				
				km : KeyUsage
				
				km : KeyOrigin
				
				km : MultiRegion
				
				km : MultiRegionKeyType
				
				km : ResourceAliases
				
				aws :ResourceTag/tag-key (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
				km : ViaService Autres conditions : km : ReEncryptOnSameKey
ReplicateKey `kms:ReplicateKey` Pour utiliser cette opération, l'appelant doit avoir les autorisations suivantes : • `kms:ReplicateKey` sur la clé principale multi-région • `kms:CreateKey` dans une politique IAM dans la région de réplica	Politique de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : ReplicaRegion

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
RetireGrant `kms:RetireGrant` L'autorisation de retirer un octroi est déterminée principalement par l'octroi. Une politique seule ne peut pas autoriser l'accès à cette opération. Pour de plus amples informations, veuillez consulter Retrait et révocation d'octrois.	Politique IAM (Cette autorisation n'est pas effective dans une politique de clé.)	Oui	Clé KMS	km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
RevokeGrant `kms:RevokeGrant`	Stratégie de clé	Oui	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions : km : GrantIsForAWSResource

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
ScheduleKeyDeletion `kms:ScheduleKeyDeletion`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
Sign (Signer) `kms:Sign`	Politique de clé	Oui	Clé KMS (asymétrique uniquement)	Conditions de signature et de vérification : km : MessageType km : RequestAlias km : SigningAlgorithm Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
TagResource `kms:TagResource`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Conditions d'étiquetage : aws :RequestTag/tag-key (clé de conditionAWS globale) aws : TagKeys (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
UntagResource `kms:UntagResource`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Conditions d'étiquetage : aws :RequestTag/tag-key (clé de conditionAWS globale) aws : TagKeys (clé de conditionAWS globale)

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
UpdateAlias `kms:UpdateAlias` Pour utiliser cette opération, l'appelant doit avoir l'autorisation `kms:UpdateAlias` sur trois ressources : • L'alias • La clé KMS actuellement associée • La clé KMS nouvellement associée Pour plus de détails, veuillez consulter Contrôle de l'accès aux alias.	Politique IAM (pour l'alias)	Non	Alias	Aucune (lorsque vous contrôlez l'accès à l'alias)
	Politique de clé (pour les clés KMS)	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService
UpdateCustomKeyStore `kms:UpdateCustomKeyStore`	Politique IAM	Non	*	km : CallerAccount

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
UpdateKeyDescription `kms:UpdateKeyDescription`	Stratégie de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
UpdatePrimaryRegion `kms:UpdatePrimaryRegion` Pour utiliser cette opération, l'appelant doit avoir l'autorisation `kms:UpdatePrimaryRegion` sur la clé principale multi-région qui deviendra une clé de réplica et sur la clé de réplica multi-région qui deviendra la clé principale.	Politique de clé	Non	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Autres conditions km : PrimaryRegion

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
Vérification `kms:Verify`	Politique de clé	Oui	Clé KMS (asymétrique uniquement)	Conditions de signature et de vérification : km : MessageType km : RequestAlias km : SigningAlgorithm Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService

Actions et autorisations	Type de stratégie	Utilisation inter-comptes	Ressources (pour les politiques IAM)	AWS KMS clés de condition
VerifyMac `kms:VerifyMac`	Stratégie de clé	Oui	Clé KMS	Conditions pour les opérations de clé KMS : km : CallerAccount km : KeySpec km : KeyUsage km : KeyOrigin km : MultiRegion km : MultiRegionKeyType km : ResourceAliases aws :ResourceTag/tag-key (clé de conditionAWS globale) km : ViaService Conditions des opérations de chiffrement : km : MacAlgorithm km : RequestAlias

Descriptions des colonnes

Les colonnes de ce tableau fournissent les informations suivantes :

- Actions et autorisations répertorie chaque opération AWS KMS d'API et l'autorisation qui autorise l'opération. Vous spécifiez l'opération dans l'élément `Action` d'une instruction de politique.

- Policy type (Type de politique) indique si l'autorisation peut être utilisée dans une politique de clé ou une politique IAM.

  Key policy (Politique de clé) signifie que vous pouvez spécifier l'autorisation dans la politique de clé. Lorsque la politique de clé contient l'instruction de politique qui active les politiques IAM, vous pouvez spécifier l'autorisation dans une politique IAM.

  IAM policy (Politique IAM) signifie que vous pouvez spécifier l'autorisation uniquement dans une politique IAM.

- Cross-account use (Utilisation inter-comptes) indique les opérations que les utilisateurs autorisés peuvent effectuer sur des ressources dans un autre Compte AWS.

  Une valeur de Yes (Oui) signifie que les principaux peuvent effectuer l'opération sur des ressources dans un autre Compte AWS.

  Une valeur de No (Non) signifie que les principaux peuvent effectuer l'opération uniquement sur des ressources dans leur propre Compte AWS.

  Si vous accordez à un principal dans un compte différent une autorisation qui ne peut pas être utilisée sur une ressource inter-comptes, l'autorisation n'est pas effective. Par exemple, si vous TagResource autorisez un mandant d'un autre compte KMS : à utiliser une clé KMS dans votre compte, ses tentatives de balisage de la clé KMS dans votre compte échoueront.

- Resources répertorie les AWS KMS ressources auxquelles les autorisations s'appliquent. AWS KMS prend en charge deux types de ressources : une clé KMS et un alias. Dans une politique de clé, la valeur de l'élément `Resource` est toujours `*`, ce qui indique la clé KMS à laquelle la politique de clé est attachée.

  Utilisez les valeurs suivantes pour représenter une AWS KMS ressource dans une politique IAM.

  Clé KMS

  Lorsque la ressource est une clé KMS, utilisez son ARN de clé. Pour obtenir de l'aide, veuillez consulter the section called “Recherche de l'ID et de l'ARN d'une clé”.

  `arn:`*`AWS_partition_name`*`:kms:`*`AWS_Region`*`:`*`AWS_account_ID`*`:key/`*`key_ID`*

  Par exemple :

  arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

Alias

Lorsque la ressource est un alias, utilisez son ARN d'alias. Pour obtenir de l'aide, veuillez consulter the section called “Recherche du nom d'alias et de l'ARN d'alias”.

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

Par exemple :

arn:aws:kms:us-west- 2:111122223333 : alias/ ExampleAlias

***** (astérisque)

Lorsque l'autorisation ne s'applique pas à une ressource particulière (clé KMS ou alias), utilisez un astérisque (*).

Dans une politique IAM pour une AWS KMS autorisation, un astérisque dans l'`Resource`élément indique toutes les AWS KMS ressources (clés KMS et alias). Vous pouvez également utiliser un astérisque dans l'`Resource`élément lorsque l' AWS KMS autorisation ne s'applique à aucune clé ou alias KMS en particulier. Par exemple, lorsque vous accordez ou refusez des autorisations `kms:CreateKey` ou `kms:ListKeys`, vous pouvez définir l'élément `Resource` sur `*` ou sur une variante spécifique au compte, telle que `arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:*`.

- AWS KMS les clés de AWS KMS condition répertorient les clés de condition que vous pouvez utiliser pour contrôler l'accès à l'opération. Vous spécifiez des conditions dans l'élément `Condition` d'une politique. Pour de plus amples informations, veuillez consulter AWS KMS clés de condition. Cette colonne inclut également les clés de conditionAWS globales qui sont prises en charge par tous les AWS services AWS KMS, mais pas par tous.

Test des autorisations

Pour utiliser AWS KMS, vous devez posséder des informations d'identification que AWS peut utiliser pour authentifier vos demandes d’API. Les informations d'identification doivent inclure des autorisations pour accéder aux clés KMS et aux alias. Les autorisations sont déterminées par les stratégies de clé, les politiques IAM, les octrois et les contrôles d'accès intercomptes. Outre le contrôle de l'accès aux clés KMS, vous pouvez contrôler l'accès à votre CloudHSM et à vos magasins de clés personnalisés.

Vous pouvez spécifier le paramètre d'API `DryRun` pour vérifier que vous disposez des autorisations nécessaires pour utiliser les clés AWS KMS. Vous pouvez également utiliser `DryRun` pour vérifier que les paramètres de demande dans un appel d'API AWS KMS sont correctement spécifiés.

Rubriques

- Quel est le DryRun paramètre ?
- Spécification DryRun à l'aide de l'API

Quel est le DryRun paramètre ?

`DryRun` est un paramètre d'API facultatif que vous spécifiez pour vérifier que les appels d'API AWS KMS aboutiront. Utilisez `DryRun` pour tester votre appel d'API, avant de passer réellement l'appel à AWS KMS. Vous pouvez modifier les valeurs suivantes :

- Que vous disposez des autorisations nécessaires pour utiliser les clés AWS KMS.
- Que vous avez correctement spécifié les paramètres lors de l'appel.

AWS KMS prend en charge l'utilisation du paramètre `DryRun` dans certaines actions d'API :

- CreateGrant
- Decrypt (Déchiffrer)
- Encrypt (Chiffrer)
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign (Signer)
- Verify (Vérifier)
- VerifyMac

L'utilisation du paramètre `DryRun` entraînera des frais et sera facturée comme une demande d'API standard. Pour plus d'informations sur la tarification AWS KMS, consultez Tarification AWS Key Management Service.

Toutes les demandes d'API utilisant le paramètre `DryRun` s'appliquent au quota de demandes de l'API et peuvent entraîner une exception de limitation si vous dépassez un quota de demandes d'API. Par exemple, le fait d'appeler Decrypt avec `DryRun` ou sans `DryRun` compte pour le même quota d'opérations cryptographiques. Pour en savoir plus, veuillez consulter Limitation des demandes AWS KMS.

Chaque appel à une opération d'API AWS KMS est capturé comme événement dans un journal AWS CloudTrail. Le résultat de toutes les opérations qui spécifient le `DryRun` paramètre apparaît dans votre CloudTrail journal. Pour plus d'informations, consultez Journalisation des appels d'API AWS KMS avec AWS CloudTrail.

Spécification DryRun à l'aide de l'API

Pour utiliser `DryRun`, spécifiez le paramètre `–dry-run` dans les commandes AWS CLI et les appels d'API AWS KMS qui prennent en charge le paramètre. Lorsque vous le ferez, AWS KMS vérifiera si votre appel aboutit. Les appels AWS KMS qui utilisent `DryRun` échoueront toujours et renverront un message contenant des informations sur le motif d'échec de l'appel. Le message peut inclure les exceptions suivantes :

- `DryRunOperationException` - La demande aboutirait si `DryRun` n'était pas spécifié.
- `ValidationException` - La demande n'a pas réussi à spécifier un paramètre d'API incorrect.
- `AccessDeniedException` - Vous ne disposez pas des autorisations pour exécuter l'action d'API spécifiée sur la ressource KMS.

Par exemple, la commande suivante utilise l'CreateGrantopération et crée une autorisation qui permet aux utilisateurs autorisés à assumer le `keyUserRole` rôle d'appeler l'opération de déchiffrement sur une clé KMS symétrique spécifiée. Le paramètre `DryRun` est spécifié.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

Clés à usage spécial

AWS Key Management Service (AWS KMS) prend en charge différents types de clés pour différentes utilisations.

Lorsque vous créez une AWS KMS key, vous obtenez par défaut une clé KMS pour le chiffrement symétrique. Dans AWS KMS, une clé KMS de chiffrement symétrique représente une clé AES-GCM 256 bits utilisée pour le chiffrement et le déchiffrement, sauf dans les régions de Chine, où elle représente une clé symétrique de 128 bits qui utilise le chiffrement SM4. Les éléments de clé symétrique ne quittent jamais AWS KMS non chiffrés. À moins que votre tâche ne nécessite explicitement des clés de chiffrement asymétriques ou HMAC, les clés KMS de chiffrement symétriques, qui ne quittent jamais AWS KMS non chiffrées, sont un bon choix. De même, les services AWS qui sont intégrés à AWS KMS utilisent des clés KMS de chiffrement symétriques pour chiffrer vos données. Ces services ne prennent pas en charge le chiffrement avec des clés KMS asymétriques.

Vous pouvez utiliser une clé KMS de chiffrement symétrique dans AWS KMS pour chiffrer, déchiffrer et rechiffrer les données, générer des clés de données et des paires de clés de données, et générer des chaînes d'octets aléatoires. Vous pouvez importer vos propres éléments de clé dans une clé KMS de chiffrement symétrique et créer des clés KMS de chiffrement symétriques dans des magasins de clés personnalisés. Pour obtenir une table comparant les opérations que vous pouvez effectuer sur des clés KMS symétriques et asymétriques, veuillez consulter Référence des types de clés.

AWS KMS prend également en charge les types de clés KMS à usage spécial suivants :

- Clés RSA asymétriques pour la cryptographie à clé publique
- Clés RSA et ECC asymétriques pour la signature et la vérification
- Clés SM2 asymétriques (régions de Chine uniquement) pour la cryptographie à clé publique ou la signature et la vérification
- Clés HMAC pour générer et vérifier les codes d'authentification de message utilisant hash
- Clés multi-région (symétriques et asymétriques) qui fonctionnent comme des copies de la même clé dans différentes Régions AWS
- Clés avec éléments de clé importés que vous fournissez
- Clés dans un magasin de clés personnalisé soutenu par un cluster AWS CloudHSM ou un gestionnaire de clés externe en dehors d'AWS.

Choix d'un type de clé KMS

AWS KMS prend en charge plusieurs types de clés KMS : clés de chiffrement symétriques, clés HMAC symétriques, clés de chiffrement asymétriques et clés de signature asymétriques.

Les clés KMS diffèrent, car elles contiennent des clés de chiffrement différentes.

- Clé KMS de chiffrement symétrique : représente une clé de chiffrement AES-GCM 256 bits, sauf dans les régions de Chine, où elle représente une clé de chiffrement SM4 128 bits. Les éléments de clé symétrique ne quittent jamais AWS KMS non chiffrés. Pour utiliser votre clé KMS de chiffrement symétrique, vous devez appeler AWS KMS.

  Les clés de chiffrement symétriques, qui sont les clés KMS par défaut, sont idéales pour la plupart des utilisations. Si vous avez besoin d'une clé KMS pour protéger vos données dans un Service AWS, utilisez une clé de chiffrement symétrique, sauf s'il vous est proposé d’utiliser un autre type de clé.
- Clé KMS asymétrique : représente une paire de clés publiques et de clés privées mathématiquement liées entre elles que vous pouvez utiliser pour chiffrer et déchiffrer ou signer et vérifier, mais pas les deux. La clé privée ne quitte jamais AWS KMS non chiffrée. Vous pouvez utiliser la clé publique dans AWS KMS en appelant des opérations d'API AWS KMS ou télécharger la clé publique et l'utiliser hors de AWS KMS.
- Clé HMAC (symétrique) : représente une clé symétrique de longueur variable utilisée pour générer et vérifier les codes d'authentification de message utilisant hash. Les éléments de clé dans une clé KMS HMAC ne quittent jamais AWS KMS non chiffrés. Pour utiliser votre clé KMS HMAC, vous devez appeler AWS KMS.

Le type de clé KMS que vous créez dépend en grande partie de la façon dont vous prévoyez d'utiliser la clé KMS, de vos exigences de sécurité et de vos exigences d'autorisation. Lors de la création de votre clé KMS, notez que la configuration de chiffrement de la clé KMS, y compris sa spécification et son utilisation, est établie lorsque vous la créez et qu'elle ne peut pas être modifiée.

Utilisez les conseils suivants pour déterminer le type de clé KMS dont vous avez besoin en fonction de votre cas d'utilisation.

Chiffrer et déchiffrer des données

Utilisez une clé KMS symétrique pour la plupart des cas d'utilisation nécessitant le chiffrement et le déchiffrement de données. L'algorithme de chiffrement symétrique qu'utilise AWS KMS est

rapide, efficace et assure la confidentialité et l'authenticité des données. Il prend en charge le chiffrement authentifié avec des données authentifiées supplémentaires (AAD), définies comme un contexte de chiffrement. Ce type de clé KMS nécessite que l'expéditeur et le destinataire des données chiffrées disposent des informations d'identification AWS valides pour appeler AWS KMS.

Si votre cas d'utilisation nécessite un chiffrement en dehors d'AWS par des utilisateurs qui ne peuvent pas appeler AWS KMS, les clés KMS asymétriques sont un bon choix. Vous pouvez distribuer la clé publique de la clé KMS asymétrique pour permettre à ces utilisateurs de chiffrer des données. Aussi, vos applications qui ont besoin de déchiffrer ces données peuvent utiliser la clé privée de la clé KMS asymétrique dans AWS KMS.

Signer des messages et vérifier des signatures

Pour signer des messages et vérifier des signatures, vous devez utiliser une clé KMS asymétrique. Vous pouvez utiliser une clé KMS avec une spécification de clé qui représente une paire de clés RSA, une paire de clés à courbe elliptique (ECC), ou une paire de clés SM2 (régions de Chine seulement). La spécification de clé que vous choisissez est déterminée par l'algorithme de signature que vous souhaitez utiliser. Plutôt que les algorithmes de signature RSA, nous vous recommandons d'utiliser les algorithmes de signature ECDSA pris en charge par les paires de clés ECC. Toutefois, vous devrez peut-être utiliser une spécification de clé et un algorithme de signature particuliers pour prendre en charge les utilisateurs qui vérifient les signatures en dehors d'AWS.

Effectuer le chiffrement d'une clé publique

Pour effectuer le chiffrement d'une clé publique, vous devez utiliser une clé KMS asymétrique avec une spécification de clé RSA ou une spécification de clé SM2 (régions de Chine seulement). Pour chiffrer des données dans AWS KMS avec la clé publique d'une paire de clés KMS, utilisez l'opération Encrypt (Chiffrer). Vous pouvez également télécharger la clé publique et la partager avec les parties qui ont besoin de chiffrer les données à l'extérieur de AWS KMS.

Lorsque vous téléchargez la clé publique d'une clé KMS asymétrique, vous pouvez l'utiliser à l'extérieur de AWS KMS. Mais elle n'est plus soumise aux contrôles de sécurité qui protègent la clé KMS dans AWS KMS. Par exemple, vous ne pouvez pas utiliser des AWS KMSpolitiques de clé ou des autorisations pour contrôler l'utilisation de la clé publique. Vous ne pouvez pas non plus contrôler si la clé est utilisée uniquement pour le chiffrement et le déchiffrement à l'aide des algorithmes de chiffrement pris en charge par AWS KMS. Pour plus d'informations, veuillez consulter Considérations spéciales pour le téléchargement de clés publiques.

Pour déchiffrer des données chiffrées avec la clé publique en dehors de AWS KMS, appelez l'opération Decrypt (Déchiffrer). L'opération `Decrypt` échoue si les données ont été chiffrées sous une clé publique à partir d'une clé KMS avec une utilisation de clé de `SIGN_VERIFY`. Elle échouera également si elle a été chiffrée à l'aide d'un algorithme qui AWS KMS ne prend pas en charge la clé de spécification que vous avez sélectionnée . Pour plus d'informations sur les spécifications clés et les algorithmes pris en charge, consultezSpécifications de clés asymétriques.

Pour éviter ces erreurs, toute personne qui utilise une clé publique en dehors de AWS KMS doit stocker la configuration de la clé. La AWS KMS console et la GetPublicKeyréponse fournissent les informations que vous devez inclure lorsque vous partagez la clé publique.

Générer et vérifier les codes HMAC

Pour générer et vérifier les codes d'authentification de message utilisant hash, utilisez une clé KMS HMAC. Lorsque vous créez une clé HMAC dans AWS KMS, AWS KMS crée et protège vos éléments de clé et garantit que vous utilisez les algorithmes MAC appropriés pour votre clé. Les codes HMAC peuvent également être utilisés comme nombres pseudo-aléatoires et dans certains scénarios pour la signature symétrique et la création de jeton.

Les clés KMS HMAC sont des clés symétriques. Lors de la création d'une clé KMS HMAC dans la console AWS KMS, choisissez le type de clé `Symmetric`.

Utilisation avec des services AWS

Pour créer une clé KMS à utiliser avec un service AWS qui est intégré à AWS KMS, veuillez consulter la documentation relative au service. Les services AWS qui chiffrent vos données nécessitent une clé KMS de chiffrement symétrique.

Outre ces considérations, les opérations de chiffrement sur des clés KMS dont les spécifications sont différentes sont soumises à des tarifs et à des quotas de demande différents. Pour plus d'informations sur la tarification AWS KMS, consultez Tarification AWS Key Management Service. Pour de plus amples informations sur les quotas de demande, veuillez consulter Quotas de demande.

Sélection de l'utilisation des clés

L'utilisation d'une clé KMS détermine si la clé KMS est utilisée pour chiffrer et déchiffrer, ou signer et vérifier les signatures, ou générer et vérifier des balises HMAC. Chaque clé KMS n'a qu'une seule utilisation de la clé. L'utilisation d'une clé KMS pour plusieurs types d'opérations rend le produit de toutes les opérations plus vulnérable aux attaques.

Comme indiqué dans la table suivante, les clés KMS de chiffrement symétriques ne peuvent être utilisées que pour chiffrer et déchiffrer. Les clés HMAC KMS ne peuvent être utilisées que pour générer et vérifier des codes HMAC. Les clés KMS de courbe elliptique (ECC) ne peuvent être utilisées que pour signer et vérifier. Vous devez prendre une décision sur l'utilisation des clés uniquement pour les clés KMS RSA.

Utilisation d'une clé valide pour les types des clés KMS

Type de clé KMS	Chiffrer et déchiffrer ENCRYPT_DECRYPT	Signer et vérifier SIGN_VERIFY	Générer et vérifier MAC GENERATE_VERIFY_MAC
Clés KMS de chiffrement symétrique	✓	⊗	⊗
Clés KMS HMAC (symétriques)	⊗	⊗	✓
Clés CMK asymétriques avec des paires de clés RSA	✓	✓	⊗
Clés CMK asymétriques avec des paires de clés ECC	⊗	✓	⊗
Clés KMS asymétriques avec paires de clés SM2 (régions de Chine uniquement)	✓	✓	⊗

Dans la console AWS KMS, vous choisissez d'abord le type de clé (symétrique ou asymétrique), puis l'utilisation de la clé. Le type de clé que vous choisissez détermine les options d'utilisation de clés affichées. L'utilisation des clés que vous choisissez détermine les spécifications de clés affichées, le cas échéant.

Pour choisir une utilisation de clé dans la console AWS KMS :

- Pour les clés KMS de chiffrement symétriques (par défaut), choisissez Encrypt and decrypt (Chiffrer et déchiffrer).
- Pour les clés KMS HMAC, choisissez Generate and verify MAC (Générer et vérifier le MAC).
- Pour les clés KMS asymétriques avec un élément de clé à courbe elliptique (ECC), choisissez Sign and verify (Signer et vérifier).
- Pour les clés KMS asymétriques avec un élément de clé RSA, choisissez Encrypt and decrypt (Chiffrer et déchiffrer) ou Sign and verify (Signer et vérifier).
- Pour les clés KMS asymétriques avec un élément de clé SM2, choisissez Encrypt and decrypt (Chiffrer et déchiffrer) ou Sign and verify (Signer et vérifier). La spécification de clé SM2 est disponible uniquement dans les régions de Chine.

Pour autoriser les principaux à créer des clés KMS uniquement pour une utilisation de clé particulière, utilisez la clé de KeyUsage condition kms :. Vous pouvez également utiliser la clé de condition `kms:KeyUsage` pour permettre aux principaux d'appeler des opérations d'API pour une clé KMS en fonction de son utilisation de clé. Par exemple, vous pouvez autoriser la désactivation d'une clé KMS uniquement si son utilisation de clé est SIGN_VERIFY.

Sélection des spécifications de la clé

Lorsque vous créez une clé KMS asymétrique ou une clé KMS HMAC, vous sélectionnez sa spécification de clé. La spécification de clé, qui est une propriété de chaque AWS KMS key, représente la configuration de chiffrement de votre clé KMS. Vous choisissez la spécification de clé lorsque vous créez la clé KMS et vous ne pouvez pas la modifier. Si vous avez sélectionné la mauvaise spécification de clé, supprimez la clé KMS et créez-en une autre.

> **Note**
>
> La spécification de clé d'une clé KMS était appelée « spécification de la clé principale du client ». Le `CustomerMasterKeySpec` paramètre de l'CreateKeyopération est obsolète. Utilisez plutôt le paramètre `KeySpec`. La réponse des DescribeKeyopérations `CreateKey` et inclut un `CustomerMasterKeySpec` membre `KeySpec` et ayant la même valeur.

La spécification de clé détermine si la clé KMS est symétrique ou asymétrique, le type d'élément dans la clé KMS et les algorithmes de chiffrement, les algorithmes de signature ou les algorithmes de code d'authentification de message (MAC) que AWS KMS prend en charge pour la clé KMS. La spécification de clé que vous choisissez est généralement déterminée par votre cas d'utilisation et vos exigences réglementaires. Cela dit, les opérations de chiffrement sur des clés KMS dont les spécifications sont différentes sont soumises à des tarifs et à des quotas différents. Pour plus d'informations sur la tarification, consultez la page AWS Key Management Service Pricing (Tarification). Pour de plus amples informations sur les quotas de demande, veuillez consulter Quotas de demande.

Pour déterminer les principales spécifications que les principaux de votre compte sont autorisés à utiliser pour les clés KMS, utilisez la clé de KeySpec condition kms :.

AWS KMS prend en charge les spécifications de clés suivantes pour les clés KMS :

Spécifications de clé de chiffrement symétrique (par défaut)

- SYMMETRIC_DEFAULT

Spécifications de clé HMAC

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

Spécifications de clés RSA (chiffrement et déchiffrement, ou signature et vérification)

- RSA_2048
- RSA_3072
- RSA_4096

Spécifications de la clé de courbe elliptique

- Paires de clés de courbe elliptique asymétriques recommandées par NIST (signature et vérification)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- Autres paires de clés asymétriques de courbe elliptique (signature et vérification)
  - ECC_SECG_P256K1 (secp256k1), couramment utilisé pour la crypto-monnaie.

Spécifications de clés SM2 (chiffrement et déchiffrement, ou signature et vérification)

- SM2 (régions de Chine uniquement)

Clés KMS asymétriques dans AWS KMS

AWS KMS prend en charge les clés KMS asymétriques qui représentent une paire de clés publique et privée mathématiquement liées RSA, à courbe elliptique (ECC) ou SM2 (régions de Chine seulement). Ces paires de clés sont générées dans des modules de sécurité matérielle AWS KMS certifiés conformément au Programme de validation des modules de chiffrement FIPS 140-2, sauf dans les régions Chine (Beijing) et Chine (Ningxia). La clé privée ne quitte jamais les modules HSM AWS KMS non chiffrés. Vous pouvez télécharger la clé publique pour la distribution et l'utiliser en dehors de AWS. Vous pouvez créer des clés KMS asymétriques pour le chiffrement et le déchiffrement, ou pour la signature et la vérification, mais pas pour les deux.

Vous pouvez créer et gérer les clés KMS asymétriques dans votre Compte AWS, notamment en définissant les politiques de clé, les politiques IAM et les octrois qui contrôlent l'accès à vos clés, en activant et désactivant les clés KMS, en créant des identifications et des alias, mais aussi en supprimant les clés KMS. De plus, vous pouvez auditer toutes les opérations qui utilisent ou gèrent vos clés KMS dans AWS dans les journaux AWS CloudTrail.

AWS KMS fournit également des paires de clés de données asymétriques conçues pour être utilisées pour la cryptographie côté client en dehors de AWS KMS. La clé privée d'une paire de clés de données asymétriques est protégée par une clé KMS de chiffrement symétrique dans AWS KMS.

Cette rubrique explique le fonctionnement des clés KMS asymétriques, leur différence avec d'autres clés KMS et comment déterminer le type de clés KMS à utiliser pour protéger vos données. Elle explique également le fonctionnement des paires de clés de données asymétriques et leur utilisation en dehors de AWS KMS.

Régions

Les clés KMS asymétriques et les paires de clés de données asymétriques sont prises en charge dans toutes les Régions AWS prises en charge par AWS KMS.

En savoir plus

- Pour créer des clés KMS asymétriques, veuillez consulter Création de clés KMS asymétriques. Pour créer des clés KMS de chiffrement symétriques, veuillez consulter Création de clés.

- Pour créer des clés KMS multi-région asymétriques, veuillez consulter Création de clés multi-régions.
- Pour savoir si une clé KMS est symétrique ou asymétrique, veuillez consulter Identification des clés KMS asymétriques.
- Pour obtenir un tableau comparatif des opérations d'API AWS KMS qui s'appliquent à chaque type de clé KMS, veuillez consulter the section called “Référence des types de clés”.
- Pour contrôler l'accès aux spécifications de clés, à l'utilisation des clés, aux algorithmes de chiffrement et aux algorithmes de signature que les principaux dans votre compte peuvent utiliser pour les clés KMS et les clés de données, veuillez consulter the section called “AWS KMS clés de condition”.
- Pour en savoir plus sur les quotas de demande qui s'appliquent aux différents types de clés KMS, veuillez consulter the section called “Quotas de demande”.
- Pour savoir comment signer des messages et vérifier des signatures à l'aide des clés KMS asymétriques, veuillez consulter Signature numérique avec la nouvelle fonction de clés asymétriques d'AWS KMS dans le blog de sécurité AWS.

Rubriques

- Clés KMS asymétriques
- Création de clés KMS asymétriques
- Téléchargement de clés publiques
- Identification des clés KMS asymétriques
- Spécifications de clés asymétriques

Clés KMS asymétriques

Vous pouvez créer une clé KMS asymétrique dans AWS KMS. Une clé KMS asymétrique représente une paire de clés publiques et de clés privées mathématiquement liées entre elles. Vous pouvez donner la clé publique à n'importe qui, même s'il ne s'agit pas d'une personne de confiance, mais la clé privée doit rester secrète.

Dans une clé KMS asymétrique, la clé privée est créée dans AWS KMS et ne quitte jamais AWS KMS non chiffrée. Pour utiliser la clé privée, vous devez appeler AWS KMS. Vous pouvez utiliser la clé publique dans AWS KMS en appelant les opérations d'API AWS KMS. Ou vous pouvez télécharger la clé publique et l'utiliser en dehors de AWS KMS.

Si votre cas d'utilisation nécessite un chiffrement en dehors d'AWS par des utilisateurs qui ne peuvent pas appeler AWS KMS, les clés KMS asymétriques sont un bon choix. Toutefois, si vous créez une clé KMS pour chiffrer les données que vous stockez ou gérez dans un service AWS, utilisez une clé KMS de chiffrement symétrique. Les services AWS qui sont intégrés à AWS KMS utilisent uniquement des clés KMS de chiffrement symétriques pour chiffrer vos données. Ces services ne prennent pas en charge le chiffrement avec des clés KMS asymétriques.

AWS KMS prend en charge trois types de clés KMS asymétriques.

- Clés KMS RSA : une clé KMS avec une paire de clés RSA pour chiffrer et déchiffrer ou pour signer et vérifier (mais pas les deux). AWS KMS prend en charge plusieurs longueurs de clés pour différentes exigences de sécurité.
- Clés KMS de courbe elliptique (ECC) : une clé KMS avec une paire de clés de courbe elliptique pour la signature et la vérification. AWS KMSprend en charge plusieurs courbes couramment utilisées.
- Clés KMS SM2 (régions de Chine uniquement) : une clé KMS avec une paire de clés SM2 pour chiffrer et déchiffrer ou pour signer et vérifier (mais pas les deux).

Pour obtenir de l'aide quant au choix de la configuration de votre clé asymétrique, veuillez consulter Choix d'un type de clé KMS. Pour plus de détails techniques sur les algorithmes de chiffrement et de signature que AWS KMS prend en charge pour les clés KMS RSA, veuillez consulter Spécifications des clés RSA. Pour plus de détails techniques sur les algorithmes de signature que AWS KMS prend en charge pour les clés KMS ECC, veuillez consulter Spécifications des clés de courbe elliptique. Pour plus de détails techniques sur les algorithmes de chiffrement et de signature que AWS KMS prend en charge pour les clés KMS SM2 (régions chinoises uniquement), consultez SM2 key spec (Spécifications des clés SM2).

Pour obtenir un tableau comparatif des opérations que vous pouvez effectuer sur les clés KMS symétriques et asymétriques, veuillez consulter Comparaison des clés KMS symétriques et asymétriques. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Régions

Les clés KMS asymétriques et les paires de clés de données asymétriques sont prises en charge dans toutes les Régions AWS prises en charge par AWS KMS.

Création de clés KMS asymétriques

Vous pouvez créer des clés KMS asymétriques dans la AWS KMS console, à l'aide de l'CreateKeyAPI ou à l'aide d'un AWS CloudFormationmodèle. Une clé KMS asymétrique représente une paire de clés publiques et privées qui peut être utilisée pour le chiffrement ou la signature. La clé privée reste dans AWS KMS. Pour télécharger la clé publique pour l'utiliser en dehors de AWS KMS, veuillez consulter Téléchargement de clés publiques.

Si vous créez une clé KMS pour chiffrer les données que vous stockez ou gérez dans un service AWS, utilisez une clé KMS de chiffrement symétrique. Les services AWS qui s'intègrent à AWS KMS ne prennent pas en charge les clés KMS asymétriques. Pour obtenir de l'aide sur la création d'une clé KMS symétrique ou asymétrique, veuillez consulter Choix d'un type de clé KMS.

Pour obtenir plus d'informations sur les autorisations nécessaires pour créer des clés KMS, consultez Autorisations de création de clés KMS.

Rubriques

- Création de clés KMS asymétriques (console)
- Création de clés KMS asymétriques (API AWS KMS)

Création de clés KMS asymétriques (console)

Vous pouvez utiliser la AWS Management Console pour créer des AWS KMS keys asymétriques (clés KMS). Chaque clé KMS asymétrique représente une paire de clés publique et privée.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.

4. Choisissez Create key.
5. Pour créer une clé KMS asymétrique, dans Key type (Type de clé), choisissez Asymmetric (Asymétrique).

   Pour de plus amples informations sur la création d'une clé KMS de chiffrement symétrique dans la console AWS KMS, veuillez consulter Création de clés KMS de chiffrement symétriques (console).
6. Pour créer une clé KMS asymétrique pour le chiffrement de clé publique, dans Key usage (Utilisation de la clé), choisissez Encrypt and decrypt (Chiffrer et déchiffrer). Ou pour créer une clé KMS asymétrique pour la signature des messages et la vérification des signatures, dans Key usage (Utilisation de la clé), choisissez Sign and verify (Signer et vérifier).

   Pour obtenir de l'aide sur le choix d'une valeur d'utilisation de clé, veuillez consulter Sélection de l'utilisation des clés.
7. Sélectionnez une spécification (Key spec (Spécifications de la clé)) pour votre clé KMS asymétrique.

   Souvent, la spécification de clé que vous sélectionnez est déterminée par des exigences réglementaires, de sécurité ou métier. Elle peut également être influencée par la taille des messages que vous devez chiffrer ou signer. En général, les clés de chiffrement plus longues résistent mieux aux attaques par force brute.

   Pour obtenir de l'aide sur le choix d'une spécification de clé, veuillez consulter Sélection des spécifications de la clé.
8. Choisissez Suivant.
9. Saisissez un alias pour la clé KMS. Le nom d'alias ne peut pas commencer par **`aws/`**. Le préfixe **`aws/`** est réservé par Amazon Web Services pour représenter Clés gérées par AWS dans votre compte.

   Un alias est un nom convivial que vous pouvez utiliser pour identifier la clé KMS dans la console et dans certaines API AWS KMS. Nous vous conseillons de choisir un alias qui indique le type de données que vous envisagez de protéger ou l'application que vous pensez utiliser avec la clé KMS.

   Les alias sont requis lorsque vous créez une clé KMS dans la AWS Management Console. Vous ne pouvez pas spécifier d'alias lorsque vous utilisez l'CreateKeyopération, mais vous pouvez

utiliser la console ou l'CreateAliasopération pour créer un alias pour une clé KMS existante. Pour plus de détails, veuillez consulter Utilisation des alias.

10. (Facultatif) Saisissez une description pour la clé KMS.

    Saisissez une description qui explique le type de données que vous envisagez de protéger ou l'application que vous pensez utiliser avec la clé KMS.

    Vous pouvez ajouter une description maintenant ou la mettre à jour à tout moment, sauf si l'état de la clé est `Pending Deletion` ou `Pending Replica Deletion`. Pour ajouter, modifier ou supprimer la description d'une clé gérée par le client existante, modifiez la description dans l'opération AWS Management Console ou utilisez l'UpdateKeyDescriptionopération.

11. (Facultatif) Saisissez une clé de balise et une valeur de balise facultative. Pour ajouter plus d'une balise à la clé KMS, sélectionnez Add tag (Ajouter une balise).

    Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour de plus amples informations sur l'étiquetage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.

12. Choisissez Suivant.

13. Sélectionnez les utilisateurs et les rôles IAM qui peuvent administrer la clé KMS.

    > **Note**
    >
    > Cette politique de clé donne au Compte AWS le contrôle total de cette clé KMS. Il permet aux administrateurs de compte d'utiliser des politiques IAM pour autoriser d'autres principals à gérer la clé KMS. Pour plus de détails, consultez the section called “politique de clé par défaut”.
    > Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

14. (Facultatif) Pour empêcher les utilisateurs et les rôles IAM sélectionnés de supprimer cette clé KMS, dans la section Key deletion (Suppression de la clé) en bas de la page, décochez la case

Allow key administrators to delete this key (Autoriser les administrateurs de clé à supprimer cette clé).

15. Choisissez Suivant.
16. Sélectionnez les utilisateurs et les rôles IAM qui peuvent utiliser la clé KMS pour les opérations cryptographiques.

> **Note**
>
> Cette politique de clé donne au Compte AWS le contrôle total de cette clé KMS. Il permet aux administrateurs de compte d'utiliser des politiques IAM pour autoriser d'autres principals à utiliser la clé KMS dans les opérations de chiffrement. Pour plus de détails, consultez the section called “politique de clé par défaut”.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

17. (Facultatif) Vous pouvez autoriser d'autres Comptes AWS à utiliser cette clé KMS pour les opérations cryptographiques. Pour cela, dans la section Autres Comptes AWS en bas de la page, sélectionnez Ajouter un autre Compte AWS et saisissez le numéro d'identification Compte AWS d'un compte externe. Pour ajouter plusieurs comptes externes, répétez cette étape.

> **Note**
>
> Pour autoriser les principaux des comptes externes à utiliser la clé KMS, les administrateurs du compte externe doivent créer des politiques IAM qui fournissent ces autorisations. Pour plus d’informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

18. Choisissez Suivant.
19. Passez en revue les paramètres de clé que vous avez choisis. Vous pouvez toujours revenir en arrière et modifier tous les paramètres.
20. Choisissez Finish (Terminer) pour créer la clé KMS.

Création de clés KMS asymétriques (API AWS KMS)

Vous pouvez utiliser cette CreateKeyopération pour créer une asymétriqueAWS KMS key. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Lorsque vous créez une clé KMS asymétrique, vous devez spécifier le paramètre `KeySpec`, qui détermine le type de clés que vous créez. En outre, vous devez spécifier une valeur `KeyUsage` ENCRYPT_DECRYPT ou SIGN_VERIFY. Vous ne pouvez pas modifier ces propriétés après la création de la clé KMS.

L'`CreateKey`opération ne vous permet pas de spécifier un alias, mais vous pouvez l'CreateAliasutiliser pour créer un alias pour votre nouvelle clé KMS.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans les champs `Description` ou `Tags`. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

L'exemple suivant utilise l'opération `CreateKey` pour créer une clé KMS asymétrique de clés RSA 4 096 bits conçues pour le chiffrement d'une clé publique.

```
$ aws kms create-key --key-spec RSA_4096 --key-usage ENCRYPT_DECRYPT
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1569973196.214,
        "MultiRegion": false,
        "KeySpec": "RSA_4096",
        "CustomerMasterKeySpec": "RSA_4096",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
```

```
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

L'exemple de commande suivant crée une clé KMS asymétrique qui représente une paire de clés ECDSA utilisées pour la signature et la vérification. Vous ne pouvez pas créer une paire de clés de courbe elliptique pour le chiffrement et le déchiffrement.

```
$ aws kms create-key --key-spec ECC_NIST_P521 --key-usage SIGN_VERIFY
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": 1570824817.837,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ],
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "AWSAccountId": "111122223333",
        "KeySpec": "ECC_NIST_P521",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Enabled": true,
        "MultiRegion": false,
        "KeyUsage": "SIGN_VERIFY"
    }
}
```

Téléchargement de clés publiques

Vous pouvez afficher, copier et télécharger la clé publique à partir d'une paire de clés KMS asymétriques à l'aide de AWS Management Console ou de l'API.AWS KMS Vous devez disposer de l'autorisation `kms:GetPublicKey` sur la clé KMS asymétrique.

Chaque paire de clés KMS asymétriques se compose d'une clé privée qui ne quitte jamais AWS KMS non chiffrée et d'une clé publique que vous pouvez télécharger et partager.

Vous pouvez partager une clé publique pour permettre à d'autres utilisateurs de chiffrer des données en dehors de AWS KMS que vous ne pouvez déchiffrer qu'avec votre clé privée. Ou pour permettre à d'autres personnes de vérifier une signature numérique en dehors de AWS KMS que vous avez générée avec votre clé privée.

Lorsque vous utilisez la clé publique dans votre clé KMS asymétrique dans AWS KMS, vous bénéficiez de l'authentification, de l'autorisation et de la journalisation qui font partie de chaque opération.AWS KMS Vous réduisez également le risque de chiffrement des données qui ne peuvent pas être déchiffrées. Ces fonctionnalités ne sont pas efficaces en dehors de.AWS KMS Pour plus d'informations, consultez.Considérations particulières pour le téléchargement de clés publiques

> **Tip**
>
> Vous recherchez des clés de données ou des clés SSH ? Cette rubrique explique comment gérer les clés asymétriques dans AWS Key Management Service, où la clé privée n'est pas exportable. Pour les paires de clés de données exportables dans lesquelles la clé privée est protégée par une clé KMS de chiffrement symétrique, voir. GenerateDataKeyPair Pour obtenir de l'aide sur le téléchargement de la clé publique associée à une instance Amazon EC2, veuillez consulter Récupération de la clé publique dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux et le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

Rubriques

- Considérations particulières pour le téléchargement de clés publiques
- Téléchargement d'une clé publique (console)
- Téléchargement d'une clé publique (AWS KMS API)

Considérations particulières pour le téléchargement de clés publiques

Pour protéger vos clés KMS, AWS KMS fournit des contrôles d'accès, un chiffrement authentifié et des journaux détaillés de chaque opération. AWS KMS vous permet également d'empêcher l'utilisation des clés KMS, temporairement ou définitivement. Enfin, les opérations AWS KMS sont conçues pour minimiser le risque de chiffrement des données qui ne peuvent pas être déchiffrées. Ces fonctionnalités ne sont pas disponibles lorsque vous utilisez des clés publiques téléchargées en dehors deAWS KMS.

Autorisation

Les politiques de clés et les politiques IAM qui contrôlent l'accès à la clé KMS dans AWS KMS n'ont aucun effet sur les opérations effectuées en dehors de AWS. Tout utilisateur qui peut obtenir la clé publique peut l'utiliser en dehors de AWS KMS, même s'il n'a pas l'autorisation de chiffrer des données ou de vérifier les signatures avec la clé KMS.

Restrictions liées à l'utilisation de la clé

Les restrictions d'utilisation de la clé ne sont pas applicables en dehors de AWS KMS. Si vous appelez l'opération Encrypt avec une clé KMS ayant une `KeyUsage` de `SIGN_VERIFY`, l'opération AWS KMS échoue. Mais si vous chiffrez des données en dehors de AWS KMS avec une clé publique d'une clé KMS avec une `KeyUsage` de `SIGN_VERIFY`, les données ne peuvent pas être déchiffrées.

Restrictions de l'algorithme

Les restrictions sur les algorithmes de chiffrement et de signature que AWS KMS prend en charge ne sont pas efficaces en dehors de AWS KMS. Si vous chiffrez des données avec la clé publique à partir d'une clé KMS en dehors de AWS KMS et utilisez un algorithme de chiffrement que AWS KMS ne prend pas en charge, les données ne peuvent pas être déchiffrées.

Désactivation et suppression des clés KMS

Les actions que vous pouvez effectuer pour empêcher l'utilisation d'une clé KMS dans une opération de chiffrement dans AWS KMS n'empêchent personne d'utiliser la clé publique en dehors de AWS KMS. Par exemple, la désactivation d'une clé KMS, la planification de la suppression d'une clé KMS, la suppression d'une clé KMS ou la suppression des éléments d'une clé KMS n'ont aucun effet sur une clé publique en dehors de AWS KMS. Si vous supprimez une clé KMS asymétrique ou supprimez ou perdez ses éléments de clé, les données que vous chiffrez avec une clé publique en dehors de AWS KMS sont irrécupérables.

Journalisation

Les journaux AWS CloudTrail qui consignent chaque opération AWS KMS, y compris la requête, la réponse, la date, l'heure et l'utilisateur autorisé, n'enregistrent pas l'utilisation de la clé publique en dehors de AWS KMS.

Vérification hors ligne avec des paires de clés SM2 (régions de Chine uniquement)

Pour vérifier une signature en dehors de AWS KMS avec une clé publique SM2, vous devez spécifier l'ID distinctif. Par défaut, AWS KMS les usages `1234567812345678` comme ID

distinctif. Pour de plus amples informations, veuillez consulter Vérification hors ligne avec des paires de clés SM2 (régions de Chine uniquement).

Téléchargement d'une clé publique (console)

Vous pouvez utiliser la AWS Management Console pour afficher, copier et télécharger la clé publique à partir d'une clé KMS asymétrique de votre Compte AWS. Pour télécharger la clé publique à partir d'une clé KMS asymétrique dans un Compte AWS différent, utilisez l'API AWS KMS.

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez l'alias ou l'ID de clé d'une clé KMS asymétrique.
5. Choisissez l'onglet Cryptographic configuration (Configuration de chiffrement). Enregistrez les valeurs des champs Spécifications de la clé, Utilisation de la clé et Algorithmes de chiffrement ou Algorithmes de signature. Vous devrez utiliser ces valeurs pour utiliser la clé publique en dehors de AWS KMS. Assurez-vous de partager ces informations lorsque vous partagez la clé publique.
6. Choisissez l'onglet Clé publique.
7. Pour copier la clé publique dans le presse-papiers, choisissez Copier. Pour télécharger la clé publique dans un fichier, choisissez Télécharger.

Téléchargement d'une clé publique (AWS KMS API)

L'GetPublicKeyopération renvoie la clé publique sous forme de clé KMS asymétrique. Elle renvoie également des informations critiques dont vous avez besoin pour utiliser correctement la clé publique en dehors de AWS KMS, y compris l'utilisation de la clé et les algorithmes de chiffrement. Veillez à enregistrer ces valeurs et à les partager chaque fois que vous partagez la clé publique.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Pour identifier une clé KMS, utilisez son ID de clé, son ARN de clé, son nom d'alias ou son ARN d'alias. Lorsque vous utilisez un nom d'alias, préfixez-le avec alias/. Pour spécifier une clé KMS dans un autre Compte AWS, vous devez utiliser son ARN de clé ou son ARN d'alias.

Avant d'exécuter cette commande, remplacez l'exemple de nom d'alias par un identifiant valide pour la clé KMS. Pour exécuter cette commande, vous devez disposer `kms:GetPublicKey` d'autorisations sur la clé KMS.

```
$ aws kms get-public-key --key-id alias/example_RSA_3072

{
    "KeySpec": "RSA_3072",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ],
    "PublicKey": "MIIBojANBgkqhkiG..."
}
```

Identification des clés KMS asymétriques

Pour déterminer si une clé KMS particulière est une clé KMS asymétrique, recherchez son type de clé ou sa spécification de clé. Vous pouvez utiliser la console AWS KMS ou l'API AWS KMS.

Certaines de ces méthodes vous montreront d'autres aspects de la configuration de chiffrement d'une clé KMS, y compris son utilisation des clés et les algorithmes de chiffrement ou de signature pris en charge par la clé KMS. Vous pouvez afficher la configuration de chiffrement d'une clé KMS existante, mais vous ne pouvez pas la modifier.

Pour obtenir des informations générales sur l'affichage des clés KMS, notamment le tri, le filtrage et le choix des colonnes pour l'affichage de la console, reportez-vous à la section Affichage de clés KMS dans la console.

Rubriques

- Recherche du type de clé dans la table de clés KMS
- Recherche du type de clé sur la page Détails
- Recherche de la spécification clé à l'aide de l'API AWS KMS

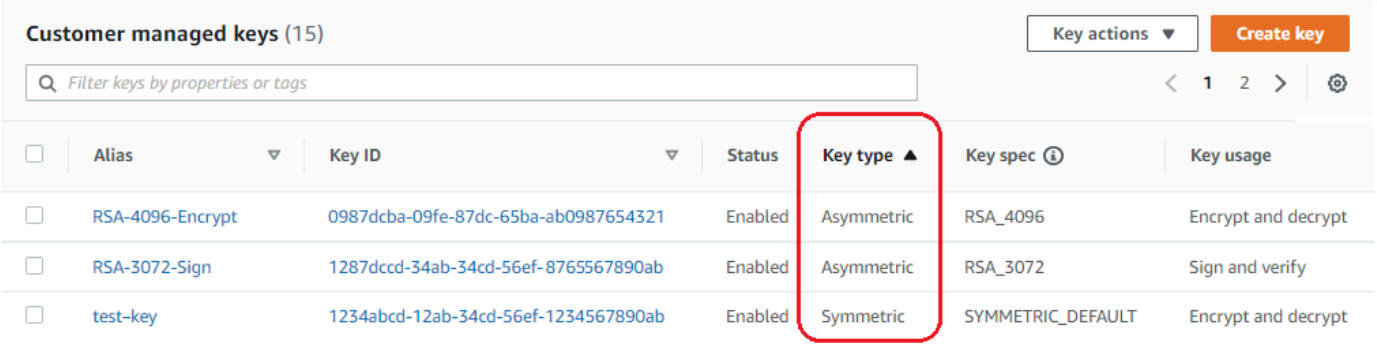
Recherche du type de clé dans la table de clés KMS

Dans la console AWS KMS, la colonne Type de clé indique si chaque clé KMS est symétrique ou asymétrique. Vous pouvez ajouter une colonne Type de clé à la table de clés KMS sur les pages Clés gérées par le client ou Clés gérées par AWS dans la console.

Pour identifier les clés KMS symétriques et asymétriques dans votre table de clés KMS, procédez comme suit.

1. Ouvrez la console AWS KMS à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de Région dans l'angle supérieur droit de la page.
3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client. Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS.
4. Les colonnes Key type (Type de clé) indiquent si chaque clé KMS est symétrique ou asymétrique. Vous pouvez également trier et filtrer selon la valeur du Key type (Type de clé).

   Si la colonne Key type (Type de clé) n'apparaît pas dans votre table de clés KMS, choisissez l'icône en forme de roue dentée dans le coin supérieur droit de la page, Key type (Type de clé), puis Confirm (Confirmer). Vous pouvez également ajouter les colonnes Key spec (Spécification de clé) et Key usage (Utilisation de clé).



Recherche du type de clé sur la page Détails

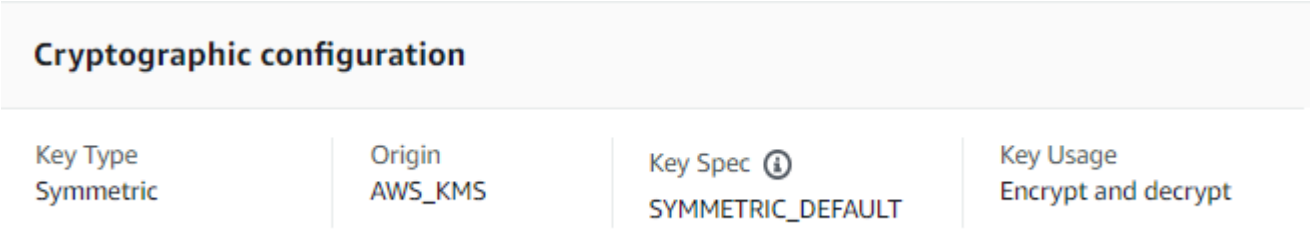
Dans la console AWS KMS, la page de détails de chaque clé KMS comprend un onglet Cryptographic Configuration (Configuration de chiffrement) qui affiche le type de clé (symétrique ou asymétrique) et d'autres détails de chiffrement sur la clé KMS.

Pour identifier les clés KMS symétriques et asymétriques sur la page de détails d'une clé KMS, procédez comme suit.

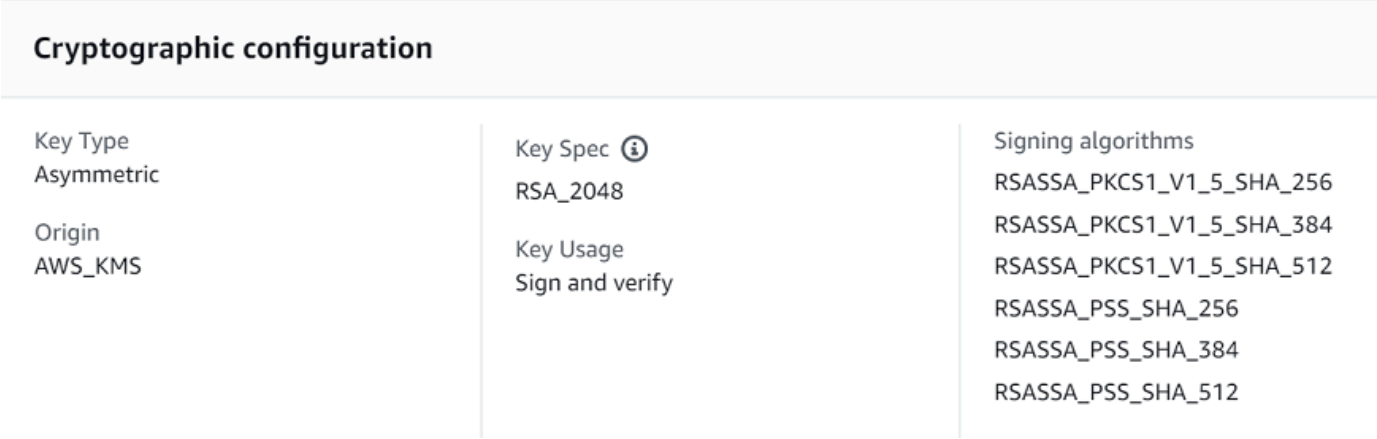
1. Ouvrez la console AWS KMS à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de Région dans l'angle supérieur droit de la page.
3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client. Pour afficher les clés de votre compte qu'AWS crée et gère pour vous, dans le panneau de navigation, choisissez Clés gérées par AWS.
4. Choisissez l'alias ou l'ID d'une clé KMS.
5. Choisissez l'onglet Cryptographic configuration (Configuration de chiffrement). Les onglets se trouvent sous la section General configuration (Configuration générale).

   L'onglet Cryptographic configuration (Configuration de chiffrement) affiche le Key Type (Type de clé), qui indique si elle est symétrique ou asymétrique. Il affiche également d'autres détails sur la clé KMS, y compris Key Usage (Utilisation de la clé), qui indique si une clé KMS peut être utilisée pour le chiffrement et le déchiffrement, ou pour la signature et la vérification. Pour les clés KMS asymétriques, il affiche les algorithmes de chiffrement ou les algorithmes de signature pris en charge par la clé KMS.

   Par exemple, voici un exemple d'onglet Cryptographic configuration (Configuration de chiffrement) pour une clé KMS de chiffrement symétrique.



   Voici un exemple d'onglet Cryptographic configuration (Configuration de chiffrement) pour une clé KMS RSA asymétrique utilisée pour la signature et la vérification.



Recherche de la spécification clé à l'aide de l'API AWS KMS

Pour déterminer si une clé KMS est symétrique ou asymétrique, utilisez l'DescribeKeyopération. Le champ `KeySpec` de la réponse contient la spécification de clé de la clé KMS. Pour une clé KMS de chiffrement symétrique, la valeur de `KeySpec` est `SYMMETRIC_DEFAULT`. Les autres valeurs indiquent une clé KMS asymétrique ou une clé KMS HMAC asymétrique.

> **Note**
>
> Le membre `CustomerMasterKeySpec` est obsolète. Utilisez à la place `KeySpec`. Pour éviter les modifications avec rupture, la réponse `DescribeKey` inclut les membres `KeySpec` et `CustomerMasterKeySpec` avec la même valeur.

Par exemple, `DescribeKey` renvoie la réponse suivante pour une clé KMS de chiffrement symétrique. La valeur `KeySpec` est `SYMMETRIC_DEFAULT`.

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
    "CreationDate": 1496966810.831,
    "Enabled": true,
    "Description": "",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
```

```
    "KeyManager": "CUSTOMER",
    "MultiRegion": false,
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

La réponse `DescribeKey` d'une clé KMS RSA asymétrique utilisée dans la signature et la vérification ressemble à cet exemple. La valeur `KeySpec` est RSA_2048 et la valeur `KeyUsage` est `SIGN_VERIFY`. L'élément `SigningAlgorithms` répertorie les algorithmes de signature valides pour la clé KMS.

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1571767572.317,
    "CustomerMasterKeySpec": "RSA_2048",
    "Enabled": false,
    "Description": "",
    "KeyState": "Disabled",
    "Origin": "AWS_KMS",
    "MultiRegion": false,
    "KeyManager": "CUSTOMER",
    "KeySpec": "RSA_2048",
    "KeyUsage": "SIGN_VERIFY",
    "SigningAlgorithms": [
        "RSASSA_PKCS1_V1_5_SHA_256",
        "RSASSA_PKCS1_V1_5_SHA_384",
        "RSASSA_PKCS1_V1_5_SHA_512",
        "RSASSA_PSS_SHA_256",
        "RSASSA_PSS_SHA_384",
        "RSASSA_PSS_SHA_512"
    ]
  }
}
```

Spécifications de clés asymétriques

Les rubriques suivantes fournissent des informations techniques sur les spécifications de clés prises en charge par AWS KMS pour les clés KMS asymétriques. Des informations sur les spécifications de clé SYMMETRIC_DEFAULT pour les clés de chiffrement symétriques sont incluses à des fins de comparaison.

Rubriques

- Spécifications de clés RSA
- Spécifications de la clé de courbe elliptique
- Spécifications de la clé SM2 (régions de Chine uniquement)
- Spécification de clé SYMMETRIC_DEFAULT

Spécifications de clés RSA

Lorsque vous utilisez une spécification de clé RSA, AWS KMS crée une clé KMS asymétrique avec une paire de clés RSA. La clé privée ne quitte jamais AWS KMS non chiffrée. Vous pouvez utiliser la clé publique dans AWS KMS, ou la télécharger pour l'utiliser en dehors de AWS KMS.

> **Warning**
>
> Lorsque vous chiffrez des données en dehors de AWS KMS, assurez-vous de pouvoir déchiffrer votre texte chiffré. Si vous utilisez la clé publique d'une clé KMS qui a été supprimée de AWS KMS, la clé publique d'une clé KMS configurée pour la signature et la vérification, ou un algorithme de chiffrement qui n'est pas pris en charge par la clé KMS, les données seront irrécupérables.

Dans AWS KMS, vous pouvez utiliser des clés KMS asymétriques avec des paires de clés RSA pour le chiffrement et le déchiffrement, ou la signature et la vérification, mais pas les deux. Cette propriété, appelée Key usage (utilisation de la clé), est déterminée en marge des spécifications de la clé, mais vous devez prendre cette décision avant de sélectionner une spécification de clé.

AWS KMS prend en charge les spécifications de clé RSA suivantes pour le chiffrement et le déchiffrement ou la signature et la vérification :

- RSA_2048

- RSA_3072
- RSA_4096

Les spécifications des clés RSA diffèrent par la longueur de la clé RSA en bits. La spécification de la clé RSA que vous choisissez peut être déterminée par vos normes de sécurité ou les exigences de votre tâche. En général, utilisez la plus grande clé qui est pratique et abordable pour votre tâche. Les opérations de chiffrement sur des clés KMS dont les spécifications de clés RSA sont différentes sont soumises à des tarifs différents. Pour de plus amples informations sur la tarification AWS KMS, veuillez consulter Tarification du service de gestion des clés AWS. Pour de plus amples informations sur les quotas de demande, veuillez consulter Quotas de demande.

Spécifications des clés RSA pour le chiffrement et le déchiffrement

Lorsqu'une clé KMS asymétrique RSA est utilisée pour le chiffrement et le déchiffrement, vous chiffrez avec la clé publique et déchiffrez avec la clé privée. Lorsque vous appelez l'opération `Encrypt` dans AWS KMS pour une clé KMS RSA, AWS KMS utilisez la clé publique dans la paire de clés RSA et l'algorithme de chiffrement que vous spécifiez pour chiffrer vos données. Pour déchiffrer le texte chiffré, appelez l'opération `Decrypt` et spécifiez les mêmes clés KMS et algorithme de chiffrement. AWS KMS utilise ensuite la clé privée dans la paire de clés RSA pour déchiffrer vos données.

Vous pouvez également télécharger la clé publique et l'utiliser pour chiffrer les données en dehors de AWS KMS. Assurez-vous d'utiliser un algorithme de chiffrement prenant AWS KMS en charge les clés KMS RSA. Pour déchiffrer le texte chiffré, appelez la fonction `Decrypt` avec les mêmes clé KMS et algorithme de chiffrement.

AWS KMS prend en charge deux algorithmes de chiffrement pour les clés KMS avec des spécifications de clé RSA. Ces algorithmes, définis dans PKCS #1 v2.2, diffèrent par la fonction de hachage qu'ils utilisent en interne. Dans AWS KMS, les algorithmes RSAES_OAEP utilisent toujours la même fonction de hachage à des fins de hachage et pour la fonction de génération de masque (MGF1). Vous devez spécifier un algorithme de chiffrement lorsque vous appelez les opérations Encrypt (Chiffrer) et Decrypt (Déchiffrer). Vous pouvez choisir un algorithme différent pour chaque requête.

Algorithmes de chiffrement pris en charge pour les spécifications de clé RSA

Algorithme de chiffrement	Description de l'algorithme
RSAES_OAEP_SHA_1	PKCS #1 v2.2, section 7.1. Chiffrement RSA avec remplissage OAEP utilisant SHA-1 pour le hachage et dans la fonction de génération de masque MGF1 avec une étiquette vide.
RSAES_OAEP_SHA_256	PKCS #1, section 7.1. Chiffrement RSA avec remplissage OAEP utilisant SHA-256 pour le hachage et dans la fonction de génération de masque MGF1 avec une étiquette vide.

Vous ne pouvez pas configurer une clé KMS pour utiliser un algorithme de chiffrement particulier. Cependant, vous pouvez utiliser la condition kms : EncryptionAlgorithm policy pour spécifier les algorithmes de chiffrement que les principaux sont autorisés à utiliser avec la clé KMS.

Pour obtenir les algorithmes de chiffrement d'une clé KMS, consultez la configuration cryptographique de la clé KMS dans la AWS KMS console ou utilisez l'DescribeKeyopération. AWS KMSfournit également les spécifications de clé et les algorithmes de chiffrement lorsque vous téléchargez votre clé publique, que ce soit dans la AWS KMS console ou à l'aide de l'GetPublicKeyopération.

Vous pouvez choisir une spécification de clé RSA en fonction de la longueur des données en texte brut que vous pouvez chiffrer dans chaque requête. Le tableau suivant indique la taille maximale, en octets, du texte brut que vous pouvez chiffrer en un seul appel à l'opération Encrypt (chiffrer). Les valeurs diffèrent selon la spécification de la clé et l'algorithme de chiffrement. Pour comparer, vous pouvez utiliser une clé KMS de chiffrement symétrique pour chiffrer jusqu'à 4 096 octets en même temps.

Pour calculer la longueur maximale du texte brut en octets pour ces algorithmes, utilisez la formule suivante : (*taille de la clé en bits* /8) - (2 * *longueur de hachage en bits*/8) - 2. Par exemple, pour RSA_2048 avec SHA-256, la taille maximale du texte brut en octets est (2048/8) - (2 * 256/8) -2 = 190.

Taille maximale du texte brut (en octets) dans une opération de chiffrement

	Algorithme de chiffrement	
Spécifications de la clé	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

Spécifications des clés RSA pour la signature et la vérification

Lorsqu'une clé KMS asymétrique RSA est utilisée pour la signature et la vérification, vous générez la signature d'un message avec la clé privée et vérifiez une signature avec la clé publique.

Lorsque vous appelez l'opération `Sign` dans AWS KMS pour une clé KMS asymétrique, AWS KMS utilise la clé privée de la paire de clés RSA, le message et l'algorithme de signature que vous spécifiez pour générer une signature. Pour vérifier la signature, appelez l'opération Vérifier. Spécifiez la signature, plus la même clé KMS, message et algorithme de signature. AWS KMS utilise ensuite la clé publique dans la paire de clés RSA pour vérifier la signature. Vous pouvez également télécharger la clé publique et l'utiliser pour vérifier la signature en dehors de AWS KMS.

AWS KMS prend en charge les algorithmes de signature suivants pour toutes les clés KMS associées à une spécification RSA. Vous devez spécifier un algorithme de signature lorsque vous appelez les opérations Sign (Signer) et Verify (Vérifier). Vous pouvez choisir un algorithme différent pour chaque requête. Lors de la signature avec des paires de clés RSA, les algorithmes RSASSA-PSS sont privilégiés. Nous incluons les algorithmes RSASSA-PKCS1-v1_5 pour des raisons de compatibilité avec les applications existantes.

Algorithmes de signature pris en charge pour les spécifications de clés RSA

Algorithme de signature	Description de l'algorithme
RSASSA_PSS_SHA_256	PKCS #1 v2.2, section 8.1, signature RSA avec remplissage PSS utilisant SHA-256 pour le résumé de message et la fonction de génération de masque MGF1 avec un salt de 256 bits

Algorithme de signature	Description de l'algorithme
RSASSA_PSS_SHA_384	PKCS #1 v2.2, section 8.1, signature RSA avec remplissage PSS utilisant SHA-384 pour le résumé des messages et la fonction de génération de masque MGF1 avec un salt de 384 bits
RSASSA_PSS_SHA_512	PKCS #1 v2.2, section 8.1, signature RSA avec remplissage PSS utilisant SHA-512 pour le résumé des messages et la fonction de génération de masque MGF1 avec un salt de 512 bits
RSASSA_PKCS1_V1_5_SHA_256	PKCS #1 v2.2, section 8.2, signature RSA avec remplissage PKCS #1v1.5 et SHA-256
RSASSA_PKCS1_V1_5_SHA_384	PKCS #1 v2.2, section 8.2, signature RSA avec remplissage PKCS #1v1.5 et SHA-384
RSASSA_PKCS1_V1_5_SHA_512	PKCS #1 v2.2, section 8.2, signature RSA avec remplissage PKCS #1v1.5 et SHA-512

Vous ne pouvez pas configurer une clé KMS pour utiliser des algorithmes de signature particuliers. Cependant, vous pouvez utiliser la condition kms : SigningAlgorithm policy pour spécifier les algorithmes de signature que les principaux sont autorisés à utiliser avec la clé KMS.

Pour obtenir les algorithmes de signature d'une clé KMS, consultez la configuration cryptographique de la clé KMS dans la AWS KMS console ou en utilisant l'DescribeKeyopération. AWS KMSfournit également les spécifications clés et les algorithmes de signature lorsque vous téléchargez votre clé publique, soit dans la AWS KMS console, soit à l'aide de l'GetPublicKeyopération.

Spécifications de la clé de courbe elliptique

Lorsque vous utilisez une spécification de clé de courbe elliptique (ECC), AWS KMS crée une clé KMS asymétrique avec une paire de clés ECC pour la signature et la vérification. La clé privée qui génère la signature ne quitte jamais AWS KMS non chiffrée. Vous pouvez utiliser la clé publique

pour vérifier les signatures dans AWS KMS, ou télécharger la clé publique pour l'utiliser en dehors de AWS KMS.

AWS KMS prend en charge les spécifications de clés ECC suivantes pour les clés KMS asymétriques.

- Paires de clés asymétriques de courbe elliptique recommandées par NIST (signature et vérification)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- Autres paires de clés asymétriques de courbe elliptique (signature et vérification)
  - ECC_SECG_P256K1 (secp256k1), couramment utilisé pour les crypto-monnaies.

La spécification de clé ECC que vous choisissez peut être déterminée par vos normes de sécurité ou les exigences de votre tâche. En général, utilisez la courbe qui est la plus pratique et abordable pour votre tâche.

Si vous créez une clé KMS asymétrique à utiliser avec les crypto-monnaies, utilisez la spécification de clé ECC_SECG_P256K1. Vous pouvez également utiliser cette spécification de clé à d'autres fins, mais elle est nécessaire pour Bitcoin et d'autres crypto-monnaies.

Les clés KMS avec des spécifications de clés ECC différentes sont tarifées différemment et sont soumises à des quotas de demande différents. Pour plus d'informations sur la tarification AWS KMS, consultez Tarification AWS Key Management Service. Pour de plus amples informations sur les quotas de demande, veuillez consulter Quotas de demande.

Le tableau suivant présente les algorithmes de signature que AWS KMS prend en charge pour chacune des spécifications de clés ECC. Vous ne pouvez pas configurer une clé KMS pour utiliser des algorithmes de signature particuliers. Cependant, vous pouvez utiliser la condition kms : SigningAlgorithm policy pour spécifier les algorithmes de signature que les principaux sont autorisés à utiliser avec la clé KMS.

Algorithmes de signature pris en charge pour les spécifications de clés ECC

Spécifications de la clé	Algorithme de signature	Description de l'algorithme
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, section 6.4, signature ECDSA utilisant la courbe spécifiée par la clé et SHA-256 pour le résumé du message.
ECC_NIST_P384	ECDSA_SHA_384	NIST FIPS 186-4, section 6.4, signature ECDSA utilisant la courbe spécifiée par la clé et SHA-384 pour le résumé du message.
ECC_NIST_P521	ECDSA_SHA_512	NIST FIPS 186-4, section 6.4, signature ECDSA utilisant la courbe spécifiée par la clé et SHA-512 pour le résumé du message.
ECC_SECG_P256K1	ECDSA_SHA_256	NIST FIPS 186-4, section 6.4, signature ECDSA utilisant la courbe spécifiée par la clé et SHA-256 pour le résumé du message.

Spécifications de la clé SM2 (régions de Chine uniquement)

La spécification clé SM2 est une spécification clé de courbe elliptique définie dans la série de spécifications GM/T publiée par le Bureau de l'administration chinoise de la cryptographie commerciale (OSCCA). La spécification de clé SM2 est disponible uniquement dans les régions de Chine. Lorsque vous utilisez une spécification de clé SM2, AWS KMS crée une clé KMS asymétrique avec une paire de clés SM2. Vous pouvez utiliser votre clé SM2 dans AWS KMS, ou télécharger la clé publique pour l'utiliser en dehors de AWS KMS.

Contrairement à la spécification de clé ECC, vous pouvez utiliser une clé SM2 KMS pour la signature et la vérification,ou le chiffrement et le déchiffrement. Vous devez spécifier l'utilisation de la clé lorsque vous créez la clé KMS, et vous ne pouvez pas la modifier une fois la clé créée.

AWS KMSprend en charge les algorithmes de cryptage et de signature SM2 suivants :

- Algorithme de chiffrement SM2PKE

  SM2PKE est un algorithme de chiffrement basé sur une courbe elliptique définie par OSCCA dans GM/T 0003.4-2012.

- Algorithme de signature SM2DSA

  SM2DSA est un algorithme de signature basé sur une courbe elliptique définie par OSCCA dans GM/T 0003.2-2012. SM2DSA nécessite un ID distinctif qui est haché avec l'algorithme de hachage SM3, puis combiné avec le message, ou résumé de message, que vous avez transmis àAWS KMS. Cette valeur concaténée est ensuite hachée et signée parAWS KMS.

Opérations hors ligne avec SM2 (régions de Chine uniquement)

Vous pouvez télécharger la clé publique de votre paire de clés SM2 pour une utilisation dans des opérations hors ligne, c'est-à-dire des opérations en dehors de AWS KMS. Toutefois, lorsque vous utiliserez votre clé publique SM2 hors ligne, vous devrez peut-être effectuer manuellement des conversions et des calculs supplémentaires. Les opérations SM2DSA peuvent vous obliger à fournir un ID distinctif ou à calculer un résumé de message. Les opérations de chiffrement SM2PKE peuvent vous obliger à convertir le texte chiffré brut en un format que AWS KMS peut accepter.

Pour vous aider dans ces opérations, la classe de `SM2OfflineOperationHelper` pour Java possède des méthodes qui exécutent les tâches à votre place. Vous pouvez utiliser cette classe d'assistance comme modèle pour d'autres fournisseurs de cryptographie.

> **Important**
>
> Le `SM2OfflineOperationHelper` code de référence est conçu pour être compatible avec Bouncy Castle version 1.68. Pour obtenir de l'aide sur les autres versions, contactezbouncycastle.org.

Vérification hors ligne avec des paires de clés SM2 (régions de Chine uniquement)

Pour vérifier une signature en dehors de AWS KMS avec une clé publique SM2, vous devez spécifier l'ID distinctif. Lorsque vous transmettez un message brut,`MessageType:RAW`, au signataire API,AWS KMSutilise l'identifiant distinctif par défaut,`1234567812345678`, défini par l'OSCCA dans GM/T 0009-2012. Vous ne pouvez pas spécifier votre propre ID distinctif dans AWS KMS.

Toutefois, si vous générez un résumé de message en dehors de AWS, vous pouvez spécifier votre propre ID distinctif, puis transmettre le résumé du message,`MessageType:DIGEST`, à AWS KMS pour signer. Pour ce faire, modifiez le paramètre `DEFAULT_DISTINGUISHING_ID` valeur dans la clé `SM2OfflineOperationHelper` classe. L'ID distinctif que vous spécifiez peut être n'importe quelle chaîne de 8 192 caractères maximum. Après que AWS KMS ait signé le résumé du message, vous aurez besoin soit du résumé du message ou du message et de l'ID distinctif utilisé pour calculer le résumé afin de le vérifier hors ligne.

classe **`SM2OfflineOperationHelper`**

Au sein de AWS KMS, les conversions de texte chiffré brut et les calculs de résumé de message SM2DSA se produisent automatiquement. Tous les fournisseurs de chiffrement ne mettent pas en œuvre SM2 de la même manière. Certaines bibliothèques, comme OpenSSL versions 1.1.1 et ultérieures, effectuent ces actions automatiquement. AWS KMS a confirmé ce comportement lors de tests avec OpenSSL version 3.0. Utilisez les classes `SM2OfflineOperationHelper` avec des bibliothèques, comme Bouncy Castle, qui vous obligent à effectuer ces conversions et ces calculs manuellement.

La classe `SM2OfflineOperationHelper` fournit des méthodes pour les opérations hors ligne suivantes :

- Calcul de l'algorithme Message Digest

  Pour générer un résumé de message hors ligne que vous pouvez utiliser pour la vérification hors ligne ou que vous pouvez transmettre à AWS KMS pour signer, utilisez la méthode `calculateSM2Digest`. La méthode `calculateSM2Digest` génère un condensé de message avec l'algorithme de hachage SM3. L'GetPublicKeyAPI renvoie votre clé publique au format binaire. Vous devez analyser la clé binaire dans un Java PublicKey. Fournissez la clé publique analysée avec le message. La méthode combine automatiquement votre message avec l'identifiant distinctif par défaut, `1234567812345678`, mais vous pouvez définir votre propre ID distinctif en modifiant la valeur `DEFAULT_DISTINGUISHING_ID`.

- Vérification

  Pour vérifier une signature hors ligne, utilisez la méthode `offlineSM2DSAVerify`. La méthode `offlineSM2DSAVerify` utilise le résumé du message calculé à partir de l'ID distinctif spécifié et du message d'origine que vous avez fourni pour vérifier la signature numérique. L'GetPublicKeyAPI renvoie votre clé publique au format binaire. Vous devez analyser la clé binaire dans un Java PublicKey. Fournissez la clé publique analysée avec le message d'origine et la signature que vous souhaitez vérifier. Pour plus d'informations, consultez .Vérification hors ligne avec des paires de clés SM2.

- Chiffrement

  Pour chiffrer du texte en clair hors ligne, utilisez la méthode `offlineSM2PKEEncrypt`. Cette méthode garantit que le texte chiffré est au format que AWS KMS peut déchiffrer. La méthode `offlineSM2PKEEncrypt` crypte le texte brut, puis convertit le texte chiffré brut produit par SM2PKE au format ASN.1. L'GetPublicKeyAPI renvoie votre clé publique au format binaire. Vous devez analyser la clé binaire dans un Java PublicKey. Fournissez la clé publique analysée avec le texte brut que vous souhaitez chiffrer.

  Si vous n'êtes pas sûr(e) de devoir effectuer la conversion, utilisez l'opération OpenSSL suivante pour tester le format de votre texte chiffré. Si l'opération échoue, vous devez convertir le texte chiffré au format ASN.1.

  ```
  openssl asn1parse -inform DER -in ciphertext.der
  ```

Par défaut, la classe `SM2OfflineOperationHelper` utilise l'ID distinctif par défaut, `1234567812345678`, lors de la génération de résumés de messages pour les opérations SM2DSA.

```
package com.amazon.kms.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.math.BigInteger;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
```

```
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.bouncycastle.crypto.CryptoException;
import org.bouncycastle.jce.interfaces.ECPublicKey;

import java.util.Arrays;

import org.bouncycastle.asn1.ASN1EncodableVector;
import org.bouncycastle.asn1.ASN1Integer;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.DERSequence;
import org.bouncycastle.asn1.gm.GMNamedCurves;
import org.bouncycastle.asn1.x9.X9ECParameters;
import org.bouncycastle.crypto.CipherParameters;
import org.bouncycastle.crypto.params.ParametersWithID;
import org.bouncycastle.crypto.params.ParametersWithRandom;
import org.bouncycastle.crypto.signers.SM2Signer;
import org.bouncycastle.jcajce.provider.asymmetric.util.ECUtil;

public class SM2OfflineOperationHelper {
    // You can change the DEFAULT_DISTINGUISHING_ID value to set your own
 distinguishing ID,
    // the DEFAULT_DISTINGUISHING_ID can be any string up to 8,192 characters long.
    private static final byte[] DEFAULT_DISTINGUISHING_ID =
 "1234567812345678".getBytes(StandardCharsets.UTF_8);
    private static final X9ECParameters SM2_X9EC_PARAMETERS =
 GMNamedCurves.getByName("sm2p256v1");

    // ***calculateSM2Digest***
    // Calculate message digest
    public static byte[] calculateSM2Digest(final PublicKey publicKey, final byte[]
 message) throws
            NoSuchProviderException, NoSuchAlgorithmException {
        final ECPublicKey ecPublicKey = (ECPublicKey) publicKey;

        // Generate SM3 hash of default distinguishing ID, 1234567812345678
        final int entlenA = DEFAULT_DISTINGUISHING_ID.length * 8;
        final byte [] entla = new byte[] { (byte) (entlenA & 0xFF00), (byte) (entlenA &
 0x00FF) };
        final byte [] a = SM2_X9EC_PARAMETERS.getCurve().getA().getEncoded();
        final byte [] b = SM2_X9EC_PARAMETERS.getCurve().getB().getEncoded();
```

```
        final byte [] xg = SM2_X9EC_PARAMETERS.getG().getXCoord().getEncoded();
        final byte [] yg = SM2_X9EC_PARAMETERS.getG().getYCoord().getEncoded();
        final byte[] xa = ecPublicKey.getQ().getXCoord().getEncoded();
        final byte[] ya = ecPublicKey.getQ().getYCoord().getEncoded();
        final byte[] za = MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(entla.length +
DEFAULT_DISTINGUISHING_ID.length + a.length + b.length + xg.length + yg.length +
                        xa.length +
ya.length).put(entla).put(DEFAULT_DISTINGUISHING_ID).put(a).put(b).put(xg).put(yg).put(xa).put
                        .array());

        // Combine hashed distinguishing ID with original message to generate final
digest
        return MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(za.length +
message.length).put(za).put(message)
                        .array());
    }

    // ***offlineSM2DSAVerify***
    // Verify digital signature with SM2 public key
    public static boolean offlineSM2DSAVerify(final PublicKey publicKey, final byte []
message,
            final byte [] signature) throws InvalidKeyException {
        final SM2Signer signer = new SM2Signer();
        CipherParameters cipherParameters =
ECUtil.generatePublicKeyParameter(publicKey);
        cipherParameters = new ParametersWithID(cipherParameters,
DEFAULT_DISTINGUISHING_ID);
        signer.init(false, cipherParameters);
        signer.update(message, 0, message.length);
        return signer.verifySignature(signature);
    }

    // ***offlineSM2PKEEncrypt***
    // Encrypt data with SM2 public key
    public static byte[] offlineSM2PKEEncrypt(final PublicKey publicKey, final byte []
plaintext) throws
            NoSuchPaddingException, NoSuchAlgorithmException, NoSuchProviderException,
InvalidKeyException,
            BadPaddingException, IllegalBlockSizeException, IOException {
        final Cipher sm2Cipher = Cipher.getInstance("SM2", "BC");
        sm2Cipher.init(Cipher.ENCRYPT_MODE, publicKey);
```

```
        // By default, Bouncy Castle returns raw ciphertext in the c1c2c3 format
        final byte [] cipherText = sm2Cipher.doFinal(plaintext);

        // Convert the raw ciphertext to the ASN.1 format before passing it to AWS KMS
        final ASN1EncodableVector asn1EncodableVector = new ASN1EncodableVector();
        final int coordinateLength = (SM2_X9EC_PARAMETERS.getCurve().getFieldSize() +
 7) / 8 * 2 + 1;
        final int sm3HashLength = 32;
        final int xCoordinateInCipherText = 33;
        final int yCoordinateInCipherText = 65;
        byte[] coords = new byte[coordinateLength];
        byte[] sm3Hash = new byte[sm3HashLength];
        byte[] remainingCipherText = new byte[cipherText.length - coordinateLength -
 sm3HashLength];

        // Split components out of the ciphertext
        System.arraycopy(cipherText, 0, coords, 0, coordinateLength);
        System.arraycopy(cipherText, cipherText.length - sm3HashLength, sm3Hash, 0,
 sm3HashLength);
        System.arraycopy(cipherText, coordinateLength, remainingCipherText,
 0,cipherText.length - coordinateLength - sm3HashLength);

        // Build standard SM2PKE ASN.1 ciphertext vector
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, 1, xCoordinateInCipherText))));
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, xCoordinateInCipherText, yCoordinateInCipherText))));
        asn1EncodableVector.add(new DEROctetString(sm3Hash));
        asn1EncodableVector.add(new DEROctetString(remainingCipherText));

        return new DERSequence(asn1EncodableVector).getEncoded("DER");
    }
}
```

Spécification de clé SYMMETRIC_DEFAULT

La spécification de clé par défaut, SYMMETRIC_DEFAULT, est la spécification de clé pour les clés KMS de chiffrement symétriques. Lorsque vous sélectionnez le type de clé Symmetric (Symétrique) et l'utilisation de clé Encrypt and decrypt (Chiffrer et déchiffrer) dans la console AWS KMS, cela sélectionne la spécification de clé `SYMMETRIC_DEFAULT`. Dans l'CreateKeyopération, si vous ne spécifiez aucune `KeySpec` valeur, SYMMETRIC_DEFAULT est sélectionné. Si vous n'avez pas de raison d'utiliser une spécification de clé différente, SYMMETRIC_DEFAULT est un bon choix.

SYMMETRIC_DEFAULT représente actuellement AES-256-GCM, un algorithme symétrique basé sur Advanced Encryption Standard (AES) en mode compteur Galois (GCM) avec des clés de 256 bits, une norme du secteur pour le chiffrement sécurisé. Le texte chiffré généré par cet algorithme prend en charge les données authentifiées supplémentaires (AAD), telles qu'un contexte de chiffrement et GCM fournit une vérification d'intégrité supplémentaire sur le texte chiffré. Pour plus d'informations techniques, veuillez consulter Détails cryptographiques AWS Key Management Service.

Les données chiffrées sous AES-256-GCM sont protégées maintenant et à l'avenir. Les cryptographes considèrent cet algorithme comme résistant quantique. Dans un avenir théorique, les attaques de calcul quantique à grande échelle sur les textes chiffrés créés sous les clés AES-GCM 256 bits réduiront la sécurité effective de la clé à 128 bits. Mais ce niveau de sécurité sera suffisant pour rendre les attaques de force brute sur les textes chiffrés AWS KMS irréalisables.

Les régions de Chine sont la seule exception, où SYMMETRIC_DEFAULT représente une clé symétrique de 128 bits qui utilise le chiffrement SM4. Vous ne pouvez créer une clé SM4 128 bits que dans les régions de Chine. Vous ne pouvez pas créer une clé AES-GCM KMS AES-GCM 256 bits dans les régions de Chine.

Vous pouvez utiliser une clé KMS de chiffrement symétrique dans AWS KMS pour chiffrer, déchiffrer et chiffrer à nouveau des données, et protéger les clés de données et paires de clés de données générées. Les services AWS qui sont intégrés à AWS KMS utilisent des clés KMS de chiffrement symétriques pour chiffrer vos données au repos. Vous pouvez importer vos propres éléments de clé dans une clé KMS de chiffrement symétrique et créer des clés KMS de chiffrement symétriques dans des magasins de clés personnalisés. Pour obtenir un tableau comparant les opérations que vous pouvez effectuer sur les clés KMS symétriques et asymétriques, veuillez consulter la Comparaison des clés KMS symétriques et asymétriques.

Pour plus de détails techniques sur AWS KMS et les clés de chiffrement symétriques, consultez AWS Key Management ServiceDétails cryptographiques.

Clés HMAC dans AWS KMS

Les clés KMS du code d'authentification de message utilisant hash (HMAC) sont des clés symétriques que vous utilisez pour générer et vérifier les HMAC dans AWS KMS. Les éléments de clé uniques associés à chaque clé KMS HMAC fournissent la clé secrète requise par les algorithmes HMAC. Vous pouvez utiliser une clé KMS HMAC avec les opérations `GenerateMac` et `VerifyMac` pour vérifier l'intégrité et l'authenticité des données dans AWS KMS.

Les algorithmes HMAC combinent une fonction de hachage cryptographique et une clé secrète partagée. Ils prennent un message et une clé secrète, comme les éléments de clé d'une clé KMS HMAC, et renvoient un code unique de taille fixe ou une balise. Si même un caractère du message change ou si la clé secrète n'est pas identique, la balise obtenue est entièrement différente. En exigeant une clé secrète, HMAC fournit également une authenticité ; il est impossible de générer une balise HMAC identique sans la clé secrète. Les HMAC sont parfois appelés signatures symétriques, car ils fonctionnent comme des signatures numériques, mais utilisent une seule clé à la fois pour la signature et la vérification.

Les clés KMS HMAC et les algorithmes HMAC qu'utilise AWS KMS sont conformes aux normes du secteur définies dans RFC 2104. L'AWS KMSGenerateMacopération génère des balises HMAC standard. Les clés KMS HMAC sont générées dans des modules de sécurité matérielle AWS KMS certifiés conformément au Programme de validation des modules de chiffrement FIPS 140-2 (sauf dans les régions Chine (Beijing) et Chine (Ningxia), et ne quittent jamais AWS KMS non chiffrées. Pour utiliser une clé KMS HMAC, vous devez appeler AWS KMS.

Vous pouvez utiliser les clés KMS HMAC pour déterminer l'authenticité d'un message, comme un jeton Web JSON (JWT), des informations de carte de crédit tokenisée ou un mot de passe envoyé. Ils peuvent également être utilisés comme fonctions de dérivation de clé (KDF) sécurisées, surtout dans les applications nécessitant des clés déterministes.

Les clés KMS HMAC offrent un avantage par rapport aux HMAC des logiciels d'application, car les éléments de clé sont générés et utilisés entièrement dans AWS KMS, sous réserve des contrôles d'accès que vous avez définis sur la clé.

> **Tip**
>
> Les bonnes pratiques recommandent de limiter la durée pendant laquelle tout mécanisme de signature, y compris un HMAC, est effectif. Cela dissuade une attaque où l'acteur utilise un message signé pour établir la validité à plusieurs reprises ou longtemps après le remplacement du message. Les balises HMAC n'incluent pas d'horodatage, mais vous pouvez inclure un horodatage dans le jeton ou le message pour vous aider à détecter le moment où il convient d'actualiser le HMAC.

Les utilisateurs autorisés peuvent créer, gérer et utiliser les clés KMS HMAC dans votre compte AWS. Cela comprend l'activation et la désactivation des clés, la définition et la modification d'alias et de balises, ainsi que la suppression de la planification de clés KMS HMAC. Vous pouvez également

contrôler l'accès aux clés KMS HMAC à l'aide de politique de clé, de politiques IAM et d'octrois. De plus, vous pouvez auditer toutes les opérations qui utilisent ou gèrent vos clés KMS HMAC dans AWS dans les journaux AWS CloudTrail. Vous pouvez créer des clés KMS HMAC avec des éléments de clé importés. Vous pouvez également créer des clés KMS multi-région HMAC qui se comportent comme des copies de la même clé KMS HMAC dans plusieurs Régions AWS.

Les clés HMAC KMS prennent uniquement en charge les opérations de chiffrement `GenerateMac` et `VerifyMac`. Vous ne pouvez pas utiliser de clés KMS HMAC pour chiffrer des données ou signer des messages, ni pour utiliser tout autre type de clé KMS dans les opérations HMAC. Lorsque vous utilisez l'opération `GenerateMac`, vous fournissez un message pouvant atteindre 4 096 octets, une clé KMS HMAC et l'algorithme MAC compatible avec la spécification de clé HMAC, tandis que `GenerateMac` calcule la balise HMAC. Pour vérifier une balise HMAC, vous devez fournir la balise HMAC, ainsi que le même message, la clé KMS HMAC et l'algorithme MAC que `GenerateMac` a utilisé pour calculer la balise HMAC d'origine. L'opération `VerifyMac` calcule la balise HMAC et vérifie qu'elle est identique à la balise HMAC fournie. Si les balises HMAC en entrée et calculées ne sont pas identiques, la vérification échoue.

Les clés KMS HMAC ne prennent pas en charge la rotation automatique des clés et vous ne pouvez pas créer de clé KMS HMAC dans un magasin de clés personnalisé.

Si vous créez une clé KMS pour chiffrer des données dans un service AWS, utilisez une clé de chiffrement symétrique. Vous ne pouvez pas utiliser de clé KMS HMAC.

Régions

Les clés KMS HMAC sont prises en charge dans toutes les Régions AWS que AWS KMS prend en charge.

En savoir plus

- Pour obtenir de l'aide sur le choix d'un type de clé KMS, veuillez consulter Choix d'un type de clé KMS.
- Pour obtenir un tableau comparatif des opérations d'API AWS KMS prises en charge par chaque type de clé KMS, veuillez consulter Référence des types de clés.
- Pour plus d'informations sur la création de clés KMS HMAC multi-région, veuillez consulter Clés multi-régions dans AWS KMS.
- Pour examiner la différence dans la politique de clé par défaut définie par la console AWS KMS pour les clés KMS HMAC, veuillez consulter the section called “Permet aux utilisateurs de clé d'utiliser la clé KMS avec les services AWS.”.

- Pour plus d'informations sur la tarification des clés KMS HMAC, veuillez consulter la tarification AWS Key Management Service.
- Pour de plus amples informations sur les quotas qui s'appliquent aux clés KMS HMAC, veuillez consulter Quotas de ressources et Quotas de demande.
- Pour plus d'informations sur la suppression de clés KMS HMAC, veuillez consulter Suppression de AWS KMS keys.
- Pour en savoir plus sur l'utilisation de HMAC pour créer des jetons web JSON, veuillez consulter Comment protéger les HMAC dans AWS KMS dans le blog de sécurité AWS.
- Écoutez le podcast :Présentation de HMAC pour AWS Key Management Service sur Le podcast AWS officiel.

Rubriques

- Spécifications de clé pour les clés KMS HMAC
- Création de clés KMS HMAC
- Contrôle de l'accès aux clés KMS HMAC
- Affichage de clés KMS HMAC

Spécifications de clé pour les clés KMS HMAC

AWS KMS prend en charge les clés HMAC symétriques de longueurs variables. La spécification de clé que vous sélectionnez peut dépendre de vos exigences de sécurité, réglementaires ou métier. La longueur de la clé détermine l'algorithme MAC utilisé dans les VerifyMacopérations GenerateMacet. En général, les clés plus longues sont plus sécurisées. Utilisez la clé la plus longue qui est pratique pour votre cas d'utilisation.

Spécification de clé HMAC	Algorithme MAC
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Création de clés KMS HMAC

Vous pouvez créer des clés KMS HMAC dans la console AWS KMS, à l'aide de l'API CreateKey ou en utilisant un modèle AWS CloudFormation.

AWS KMS prend en charge plusieurs spécifications de la clé pour les clés KMS HMAC.
La spécification de clé que vous sélectionnez pourrait être déterminée par des exigences réglementaires, de sécurité ou métier. En général, les clés plus longues résistent mieux aux attaques par force brute.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

Si vous créez une clé KMS pour chiffrer les données dans un service AWS, utilisez une clé KMS de chiffrement symétrique. Les services AWS qui s'intègrent à AWS KMS ne prennent pas en charge les clés KMS asymétriques ou les clés KMS HMAC. Pour obtenir de l'aide sur la création d'une clé KMS de chiffrement symétrique, veuillez consulter Création de clés.

En savoir plus

- Pour déterminer le type de clé KMS à créer, veuillez consulter Choix d'un type de clé KMS.
- Vous pouvez utiliser les procédures décrites dans cette rubrique pour créer une clé KMS HMAC principale multi-régions. Pour répliquer une clé HMAC multi-région, veuillez consulter the section called “Création de clés de réplica”.
- Pour obtenir plus d'informations sur les autorisations nécessaires pour créer des clés KMS, consultez Autorisations de création de clés KMS.
- Pour plus d'informations sur l'utilisation d'un AWS CloudFormation modèle pour créer une clé HMAC KMS, consultez AWS::KMS::Keyle guide de l'AWS CloudFormationutilisateur.

Rubriques

- Création de clés KMS HMAC (console)
- Création de clés KMS HMAC (API AWS KMS)

Création de clés KMS HMAC (console)

Vous pouvez utiliser la AWS Management Console pour créer des clés KMS HMAC. Les clés KMS HMAC sont des clés symétriques avec une utilisation de clé de Generate and verify MAC (Générer et vérifier le MAC). Vous pouvez également créer des clés HMAC multi-région.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key.
5. Pour Type de clé, choisissez Symétrique.

   Les clés KMS HMAC sont symétriques. Vous utilisez la même clé pour générer et vérifier les balises HMAC.
6. Pour Key usage (Utilisation de la clé), choisissez Generate and verify MAC (Générer et vérifier le MAC).

   Générer et vérifier le MAC est la seule utilisation valide de clés KMS HMAC.

   > **Note**
   >
   > Key usage (Utilisation de la clé) est affiché pour les clés symétriques uniquement lorsque les clés KMS HMAC sont prises en charge dans votre région sélectionnée.
7. Sélectionnez une spécification (Key spec [Spécifications de clé]) pour votre clé KMS HMAC.

   La spécification de clé que vous sélectionnez peut être déterminée par des exigences réglementaires, de sécurité ou métier. En général, les clés plus longues sont plus sécurisées.
8. Pour créer une clé HMAC principale multi-région, dans Advanced options (Options avancées), choisissez Multi-Region key (Clé multi-région). Les propriétés partagées que vous définissez pour cette clé KMS, comme son type de clé et son utilisation de clé, seront partagés avec ses clés de réplica. Pour plus de détails, consultez Création de clés multi-régions.

   Vous ne pouvez pas utiliser cette procédure pour créer une clé de réplica. Pour créer une clé HMAC de réplica multi-région, suivez les instructions de création d'une clé de réplica.

9. Choisissez Suivant.
10. Saisissez un alias pour la clé KMS. Le nom d'alias ne peut pas commencer par **`aws/`**. Le préfixe **`aws/`** est réservé par Amazon Web Services pour représenter Clés gérées par AWS dans votre compte.

    Nous vous recommandons d'utiliser un alias qui identifie la clé KMS en tant que clé HMAC, comme `HMAC/test-key`. Cela vous permettra d'identifier plus facilement vos clés HMAC dans la console AWS KMS où vous pouvez trier et filtrer les clés par balises et alias, mais pas par spécification de clé ou par utilisation de clé.

    Les alias sont requis lorsque vous créez une clé KMS dans la AWS Management Console. Vous ne pouvez pas spécifier d'alias lorsque vous utilisez l'CreateKeyopération, mais vous pouvez utiliser la console ou l'CreateAliasopération pour créer un alias pour une clé KMS existante. Pour plus de détails, consultez Utilisation des alias.
11. (Facultatif) Saisissez une description pour la clé KMS.

    Saisissez une description qui explique le type de données que vous envisagez de protéger ou l'application que vous pensez utiliser avec la clé KMS.

    Vous pouvez ajouter une description maintenant ou la mettre à jour à tout moment, sauf si l'état de la clé est `Pending Deletion` ou `Pending Replica Deletion`. Pour ajouter, modifier ou supprimer la description d'une clé gérée par le client existante, modifiez la description dans l'opération AWS Management Console ou utilisez l'UpdateKeyDescriptionopération.
12. (Facultatif) Saisissez une clé de balise et une valeur de balise facultative. Pour ajouter plus d'une balise à la clé KMS, sélectionnez Add tag (Ajouter une balise).

    Envisagez d'ajouter une balise qui identifie la clé en tant que clé HMAC, comme `Type=HMAC`. Cela vous permettra d'identifier plus facilement vos clés HMAC dans la console AWS KMS où vous pouvez trier et filtrer les clés par balises et alias, mais pas par spécification de clé ou par utilisation de clé.

    Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour de plus amples informations sur l'étiquetage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.
13. Choisissez Suivant.

14. Sélectionnez les utilisateurs et les rôles IAM qui peuvent administrer la clé KMS.

> **Note**
>
> Cette politique de clé donne au Compte AWS le contrôle total de cette clé KMS. Il permet aux administrateurs de compte d'utiliser des politiques IAM pour autoriser d'autres principals à gérer la clé KMS. Pour plus de détails, consultez the section called “politique de clé par défaut”.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

15. (Facultatif) Pour empêcher les utilisateurs et les rôles IAM sélectionnés de supprimer cette clé KMS, dans la section Key deletion (Suppression de la clé) en bas de la page, décochez la case Allow key administrators to delete this key (Autoriser les administrateurs de clé à supprimer cette clé).
16. Choisissez Suivant.
17. Sélectionnez les utilisateurs et les rôles IAM qui peuvent utiliser la clé KMS pour les opérations crytographiques.

> **Note**
>
> Cette politique de clé donne au Compte AWS le contrôle total de cette clé KMS. Il permet aux administrateurs de compte d'utiliser des politiques IAM pour autoriser d'autres principals à utiliser la clé KMS dans les opérations de chiffrement. Pour plus de détails, consultez the section called “politique de clé par défaut”.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

18. (Facultatif) Vous pouvez autoriser d'autres Comptes AWS à utiliser cette clé KMS pour les opérations cryptographiques. Pour cela, dans la section Autres Comptes AWS en bas de la page, sélectionnez Ajouter un autre Compte AWS et saisissez le numéro d'identification

Compte AWS d'un compte externe. Pour ajouter plusieurs comptes externes, répétez cette étape.

> **Note**
>
> Pour autoriser les principaux des comptes externes à utiliser la clé KMS, les administrateurs du compte externe doivent créer des politiques IAM qui fournissent ces autorisations. Pour plus d'informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

19. Choisissez Suivant.
20. Passez en revue les paramètres de clé que vous avez choisis. Vous pouvez toujours revenir en arrière et modifier tous les paramètres.
21. Choisissez Finish (Terminer) pour créer la clé KMS HMAC.

Création de clés KMS HMAC (API AWS KMS)

Vous pouvez utiliser cette CreateKeyopération pour créer une clé HMAC KMS. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Lorsque vous créez une clé KMS HMAC, vous devez spécifier le paramètre `KeySpec`, qui détermine le type de clé KMS. Vous devez également spécifier une valeur `KeyUsage` de GENERATE_VERIFY_MAC, même s'il s'agit de la seule valeur d'utilisation de clé valide pour les clés HMAC. Pour créer une clé multi-région KMS HMAC, ajoutez le paramètre `MultiRegion` avec la valeur `true`. Vous ne pouvez pas modifier ces propriétés après la création de la clé KMS.

L'`CreateKey`opération ne vous permet pas de spécifier un alias, mais vous pouvez l'CreateAliasutiliser pour créer un alias pour votre nouvelle clé KMS. Nous vous recommandons d'utiliser un alias qui identifie la clé KMS en tant que clé HMAC, tel que `HMAC/test-key`. Cela vous permettra d'identifier plus facilement vos clés HMAC dans la console AWS KMS où vous pouvez trier et filtrer les clés par alias, mais pas par spécification de clé ou par utilisation de clé.

Si vous essayez de créer une clé KMS HMAC dans une Région AWS dans laquelle les clés HMAC ne sont pas prises en charge, l'opération `CreateKey` retourne une `UnsupportedOperationException`.

L'exemple suivant utilise l'opération `CreateKey` pour créer une clé KMS HMAC de 512 bits.

```
$ aws kms create-key --key-spec HMAC_512 --key-usage GENERATE_VERIFY_MAC
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1669973196.214,
        "MultiRegion": false,
        "KeySpec": "HMAC_512",
        "CustomerMasterKeySpec": "HMAC_512",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_512"
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

Contrôle de l'accès aux clés KMS HMAC

Pour contrôler l'accès à une clé KMS HMAC, vous utilisez une politique de clé, requise pour chaque clé KMS. Vous pouvez également utiliser des politiques IAM et des octrois.

La politique de clé par défaut pour les clés HMAC créées dans la console AWS KMS donne aux utilisateurs de clé l'autorisation d'appeler les opérations `GenerateMac` et `VerifyMac`. Toutefois, il ne tient pas compte de l'instruction de politique de clé visant à utiliser des octrois avec des services AWS. Si vous créez des clés HMAC à l'aide de l'opération `CreateKey`, vous devez spécifier ces autorisations dans la politique de clé ou dans une politique IAM.

Vous pouvez utiliser des clés de condition globales AWS et des clés de condition AWS KMS pour affiner et limiter les autorisations aux clés HMAC. Par exemple, vous pouvez utiliser la clé de condition `kms:ResourceAliases` pour contrôler l'accès aux opérations AWS KMS basées sur les alias associés à une clé HMAC. Les conditions de politique AWS KMS suivantes sont utiles pour les politiques relatives aux clés HMAC.

- Utilisation d'une clé de condition `kms:MacAlgorithm` pour limiter les algorithmes que les principaux peuvent demander lorsqu'ils appellent les opérations `GenerateMac` et `VerifyMac`. Par exemple, vous pouvez autoriser les principaux à appeler les opérations `GenerateMac`, mais uniquement lorsque l'algorithme MAC de la demande est `HMAC_SHA_384`.
- Utilisation d'une clé de condition `kms:KeySpec` pour autoriser ou empêcher les principaux de créer certains types de clés HMAC. Par exemple, pour autoriser les principaux à créer uniquement des clés HMAC, vous pouvez autoriser l'CreateKeyopération, mais utiliser la `kms:KeySpec` condition pour n'autoriser que les clés dotées d'une spécification de `HMAC_384` clé.

  Vous pouvez également utiliser la clé de condition `kms:KeySpec` pour contrôler l'accès aux autres opérations sur une clé KMS en fonction de la spécification de clé. Par exemple, vous pouvez autoriser les principaux à planifier et à annuler la suppression de clés uniquement sur les clés KMS avec une spécification de clé `HMAC_256`.
- Utilisation de la clé de condition `kms:KeyUsage` pour autoriser ou empêcher les principaux de créer des clés HMAC. Par exemple, pour autoriser les principaux à créer uniquement des clés HMAC, vous pouvez autoriser l'CreateKeyopération, mais utiliser la `kms:KeyUsage` condition pour autoriser uniquement les clés utilisant une `GENERATE_VERIFY_MAC` clé.

  Vous pouvez également utiliser la clé de condition `kms:KeyUsage` pour contrôler l'accès aux autres opérations sur une clé KMS en fonction de l'utilisation de la clé. Par exemple, vous pouvez autoriser les principaux à activer et à désactiver uniquement sur les clés KMS avec une utilisation de la clé `GENERATE_VERIFY_MAC`.

Vous pouvez également créer des octrois pour les opérations `GenerateMac` et `VerifyMac`, qui sont des opérations d'octroi. Toutefois, vous ne pouvez pas utiliser une contrainte d'octroi de contexte de chiffrement dans un octroi pour une clé HMAC. Le format de balise HMAC ne prend pas en charge les valeurs du contexte de chiffrement.

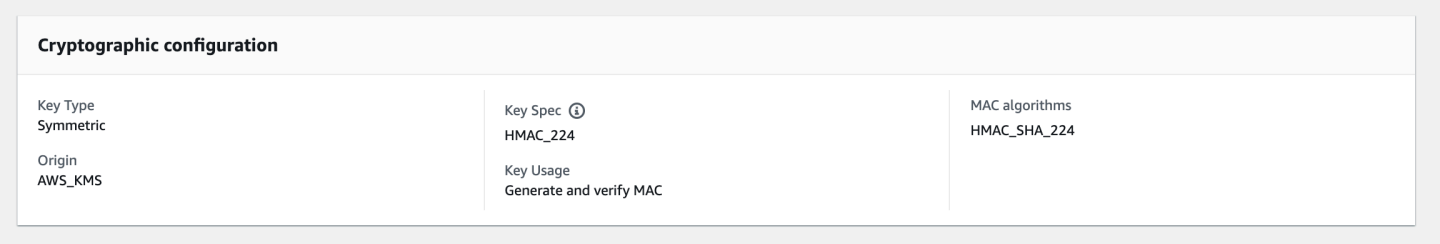
Affichage de clés KMS HMAC

Vous pouvez afficher les clés KMS HMAC dans la console AWS KMS ou en utilisant l'API DescribeKey. Vous pouvez surveiller l'utilisation de vos clés HMAC KMS dans les AWS CloudTrailjournaux et sur Amazon CloudWatch. Pour obtenir des instructions de base sur l'affichage des clés KMS, veuillez consulter Affichage des clés.

Vous pouvez distinguer les clés KMS HMAC des autres types de clés KMS par leur spécification de clé, qui commence par `HMAC`, ou leur utilisation de clé, qui est toujours Generate and verify MAC (Générer et vérifier le MAC) (`GENERATE_VERIFY_MAC`).

Les clés KMS HMAC sont incluses dans le tableau sur la page Customer managed keys (Clés gérées par le client) de la console AWS KMS. Toutefois, vous ne pouvez pas trier ou filtrer des clés KMS par spécification de clé ou utilisation de clé. Pour faciliter la recherche de vos clés HMAC, attribuez-leur un alias distinctif ou une balise distinctive. Vous pouvez ensuite trier ou filtrer par alias ou balise.

Dans la page de détails de clé pour une clé KMS HMAC, vous pouvez trouver les détails de sa configuration sur l'onglet Cryptographic configuration (Configuration de chiffrement).



Clés multi-régions dans AWS KMS

AWS KMS prend en charge des clés multi-région, qui sont des AWS KMS keys dans différentes Régions AWS pouvant être utilisées de façon interchangeable, comme si vous aviez la même clé dans plusieurs régions. Chaque ensemble de clés multi-région liées est doté des mêmes éléments de clé et ID de clé, afin que vous puissiez chiffrer des données dans une Région AWS et les déchiffrer dans une Région AWS différente, sans avoir à rechiffrer ou à procéder à un appel inter-régions vers AWS KMS.

Comme toutes les clés KMS, les clés multi-région ne quittent jamais AWS KMS non chiffrées. Vous pouvez créer des clés multi-région symétriques ou asymétriques pour le chiffrement ou la signature, créer des clés multi-régions HMAC pour la génération et la vérification de balises HMAC, mais aussi créer des clés multi-région avec des éléments de clé importés ou des éléments de clé générés par AWS KMS. Vous devez gérer chaque clé multi-région indépendamment, notamment en créant des alias et des balises, en définissant les politiques et les octrois de clé, en les activant et en les désactivant de manière sélective. Vous pouvez utiliser des clés multi-région dans le cadre de toutes les opérations de chiffrement que vous pouvez effectuer avec des clés à région unique.

Les clés multi-région sont une solution flexible et puissante pour de nombreux scénarios courants liés à la sécurité des données.

Reprise après sinistre

Dans une architecture de sauvegarde et de reprise, les clés multi-région vous permettent de traiter des données chiffrées sans interruption, même en cas de panne de Région AWS. Les données conservées dans les régions de sauvegarde peuvent être déchiffrées dans la région de sauvegarde, et les données nouvellement chiffrées dans la région de sauvegarde peuvent être déchiffrées dans la région principale lorsque cette région est restaurée.

Gestion globale des données

Les entreprises qui opèrent à l'échelle mondiale ont besoin de données distribuées dans le monde entier et qui soient disponibles entre les Régions AWS. Vous pouvez créer des clés multi-région dans toutes les régions où résident vos données, puis utiliser les clés comme s'il s'agissait d'une clé à région unique sans la latence d'un appel inter-régions ou le coût du re-chiffrement des données sous une clé différente dans chaque région.

Applications de signature distribuées

Les applications qui nécessitent des fonctionnalités de signature inter-régions peuvent utiliser des clés de signature asymétriques multi-région pour générer des signatures numériques identiques de manière cohérente et répétée dans différentes Régions AWS.

Si vous utilisez le chaînage de certificats avec un seul magasin d'approbation global (pour une seule autorité de certification [CA] racine), et des CA intermédiaires régionales signées par la CA racine, vous n'avez pas besoin de clés multi-régions. Toutefois, si votre système ne prend pas en charge les CA intermédiaires, telles que la signature d'applications, vous pouvez utiliser des clés multi-région pour assurer la cohérence des certifications régionales.

Applications active/active couvrant plusieurs régions

Certaines charges de travail et applications peuvent couvrir plusieurs régions dans des architectures active/active. Pour ces applications, les clés multi-région peuvent réduire la complexité en fournissant les mêmes éléments de clé pour les opérations simultanées de chiffrement et de déchiffrement sur les données susceptibles de se déplacer au-delà des limites de la région.

Vous pouvez utiliser des clés multi-région avec des bibliothèques de chiffrement côté client, telles que le AWS Encryption SDK, le client de chiffrement DynamoDB, et le chiffrement côté client Amazon S3. Pour obtenir un exemple d'utilisation de clés multi-région avec les tables globales Amazon DynamoDB et le client de chiffrement DynamoDB, veuillez consulter Chiffrer les données globales côté client avec les clés AWS KMS multi-région dans le blog AWS sur la sécurité.

Les services AWS qui s'intègrent à AWS KMS pour le chiffrement au repos ou les signatures numériques traitent actuellement les clés multi-région comme s'il s'agissait de clés à région unique. Ils peuvent ré-encapsulper ou re-chiffrer les données déplacées entre les régions. Par exemple, la réplication inter-régions Amazon S3 déchiffre et rechiffre les données sous une clé KMS dans la région de destination, même lors de la réplication d'objets protégés par une clé multi-région.

Les clés multi-région ne sont pas globales. Vous créez une clé principale multi-région, puis vous la répliquez dans les régions que vous sélectionnez dans une partition AWS. Vous gérez ensuite la clé multi-région dans chaque région de manière indépendante. Ni AWS ni AWS KMS ne créent ou ne répliquent automatiquement des clés multi-région dans n'importe quelle région en votre nom. Les Clés gérées par AWS, les clés KMS que les services AWS créent dans votre compte pour vous, sont toujours des clés à région unique.

Vous ne pouvez pas transformer une clé à région unique en clé multi-région. Cette conception garantit que toutes les données protégées avec les clés à région unique existantes conservent les mêmes propriétés de résidence et de souveraineté des données.

Pour la plupart des besoins en matière de sécurité des données, l'isolement régional et la tolérance aux pannes des ressources régionales font des clés à région unique AWS KMS standard la solution la plus adaptée. Toutefois, lorsque vous avez besoin de chiffrer ou de signer des données dans des applications côté client entre plusieurs régions, les clés multi-région peuvent être la solution.

Régions

Les clés multi-région sont prises en charge dans toutes les Régions AWS prises en charge par AWS KMS, à l'exception des régions Chine (Beijing) et Chine (Ningxia).

Tarification et quotas

Chaque clé d'un ensemble de clés multi-région associées compte comme une clé KMS pour la tarification et les quotas. Les quotas AWS KMS sont calculés séparément pour chaque région d'un compte. L'utilisation et la gestion des clés multi-région dans chaque région sont prises en compte dans les quotas pour cette région.

Types de clés KMS non pris en charge

Vous pouvez créer les types suivants de clés KMS multirégions :

- Clés KMS de chiffrement symétrique

- Clés KMS asymétriques
- Clés KMS HMAC
- Clés KMS avec des éléments de clé importés

Vous ne pouvez pas créer de clés multi-région dans un magasin de clés personnalisé.

Rubriques

- Contrôle de l'accès aux clés multi-région
- Création de clés multi-région
- Affichage des clés multi-région
- Gestion des clés multi-région
- Importation des éléments de clé dans des clés multi-région
- Suppression de clés multi-région

Considérations sur la sécurité pour les clés multi-région

Utilisez une clé multi-région AWS KMS uniquement lorsque vous en avez besoin. Les clés multi-région fournissent une solution flexible et évolutive pour les applications qui déplacent des données chiffrées entre Régions AWS ou ont besoin d'un accès inter-régions. Envisagez une clé multi-région si vous devez partager, déplacer ou sauvegarder des données protégées entre les régions ou si vous avez besoin de créer des signatures numériques identiques d'applications fonctionnant dans différentes régions.

Toutefois, le processus de création d'une clé multi-région déplace vos éléments de clé au-delà des frontières des Région AWS au sein de AWS KMS. Le texte chiffré généré par une clé multi-région peut potentiellement être déchiffré par plusieurs clés associées dans plusieurs emplacements géographiques. Il y a également des avantages importants pour les services et les ressources isolés dans la région. Chaque Région AWS est indépendante et isolée des autres régions. Les régions fournissent une tolérance aux pannes, une stabilité et une résilience, et peuvent également réduire la latence. Elles vous permettent de créer des ressources redondantes qui restent disponibles et qui ne sont pas affectées par les pannes dans les autres régions. Dans AWS KMS, elles veillent également à ce que chaque texte chiffré puisse être déchiffré par une seule clé.

Les clés multi-région soulèvent également de nouvelles considérations de sécurité :

- Le contrôle de l'accès et l'application de la politique de sécurité des données sont plus complexes avec les clés multi-région. Vous devez vous assurer que la politique est auditée de manière cohérente sur la clé dans plusieurs régions isolées. Vous devez également utiliser la politique pour appliquer les frontières, au lieu de vous appuyer sur des clés séparées.

  Par exemple, vous devez définir des conditions de politique sur les données pour empêcher les équipes de paie d'une région de lire les données de paie pour une autre région. En outre, vous devez utiliser le contrôle d'accès pour empêcher un scénario dans lequel une clé multi-région dans une région protège les données d'un locataire et une clé multi-région associée dans une autre région protège les données d'un autre locataire.
- L'audit des clés entre les régions est également plus complexe. Avec les clés multi-région, vous devez examiner et réconcilier les activités d'audit entre plusieurs régions afin d'obtenir une compréhension complète des activités de clé liées aux données protégées.
- La conformité aux mandats de résidence des données peut être plus complexe. Avec les régions isolées, vous pouvez garantir la résidence des données et la conformité à la souveraineté des données. Les clés KMS d'une région donnée peuvent déchiffrer des données sensibles uniquement dans cette région. Les données chiffrées dans une région peuvent rester complètement protégées et inaccessibles dans toute autre région.

  Pour vérifier la résidence des données et la souveraineté des données à l'aide de clés multi-région, vous devez implémenter des politiques d'accès et compiler des événements AWS CloudTrail dans plusieurs régions.

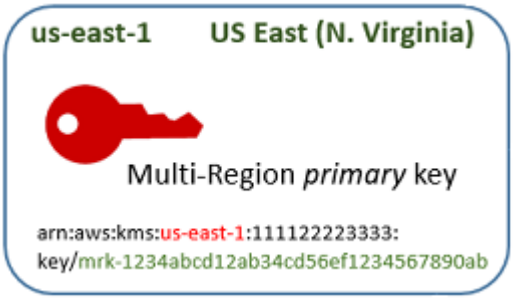
Pour faciliter la gestion du contrôle d'accès sur les clés multirégionales, l'autorisation de répliquer une clé multirégionale (kms : ReplicateKey) est distincte de l'autorisation standard de création de clés (kms :). CreateKey De plus, AWS KMS prend en charge plusieurs conditions de politique pour les clés multi-région, notamment `kms:MultiRegion`, qui autorise ou refuse l'autorisation de créer, d'utiliser ou de gérer des clés multi-région et `kms:ReplicaRegion`, qui limite les régions dans lesquelles une clé multi-région peut être répliquée. Pour plus de détails, veuillez consulter Contrôle de l'accès aux clés multi-régions.

Fonctionnement des clés multi-région

Vous commencez par créer une clé principale multi-région symétrique ou asymétrique dans une Région AWS prise en charge par AWS KMS, telle que USA Est (Virginie du Nord). Vous décidez si une clé est à région unique ou multi-région uniquement lorsque vous la créez ; vous ne pouvez pas modifier cette propriété ultérieurement. Comme pour toute clé KMS, vous définissez une politique de

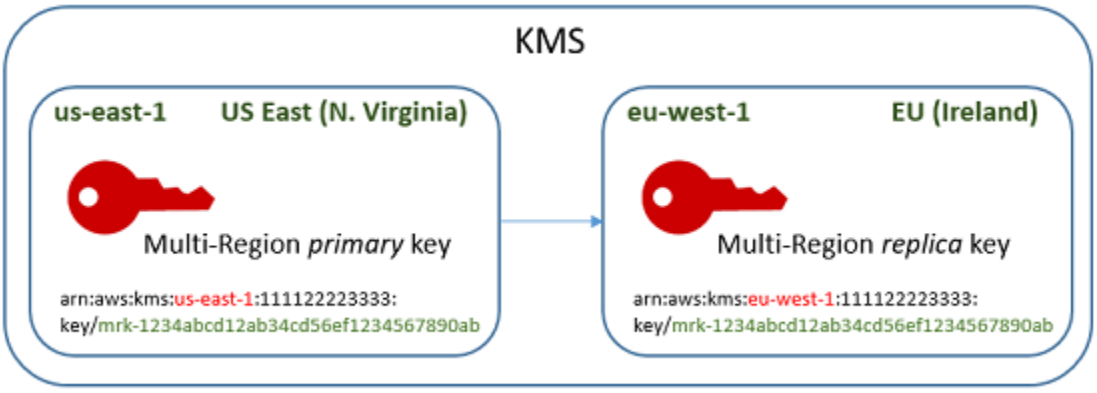
clé pour la clé multi-région, et vous pouvez créer des octrois et ajouter des alias et des balises pour la catégorisation et l'autorisation. (Ce sont des propriétés indépendantes qui ne sont pas partagées ou synchronisées avec d'autres clés.) Vous pouvez utiliser votre clé principale multi-région dans les opérations de chiffrement pour le chiffrement ou la signature.

Vous pouvez créer une clé primaire multirégionale dans la AWS KMS console ou en utilisant l'CreateKeyAPI avec le `MultiRegion` paramètre défini sur. `true` Notez que les clés multi-région ont un ID de clé distinctif qui commence par `mrk-`. Vous pouvez utiliser le préfixe `mrk-` pour identifier les clés multi-région par programmation.



Si vous le souhaitez, vous pouvez répliquer la clé principale multi-région dans une ou plusieurs Régions AWS différentes dans la même partition AWS, telle que la région UE (Irlande). Dans ce cas, AWS KMS créer une clé de réplica dans la région spécifiée, avec le même ID de clé et d'autres propriétés partagées en tant que clé principale. Ensuite, il transporte en toute sécurité les éléments de clé au-delà des limites de la région et les associe à la nouvelle clé KMS dans la région de destination, le tout dans AWS KMS. Le résultat donne deux clés multi-région associées : une clé principale et une clé de réplica, pouvant être utilisées de manière interchangeable.

Vous pouvez créer une réplique de clé multirégionale dans la AWS KMS console ou à l'aide de l'ReplicateKeyAPI.



La clé de réplica multi-région qui en résulte est une clé KMS pleinement fonctionnelle, avec les mêmes propriétés partagées que clé principale. À tous autres égards, il s'agit d'une clé KMS

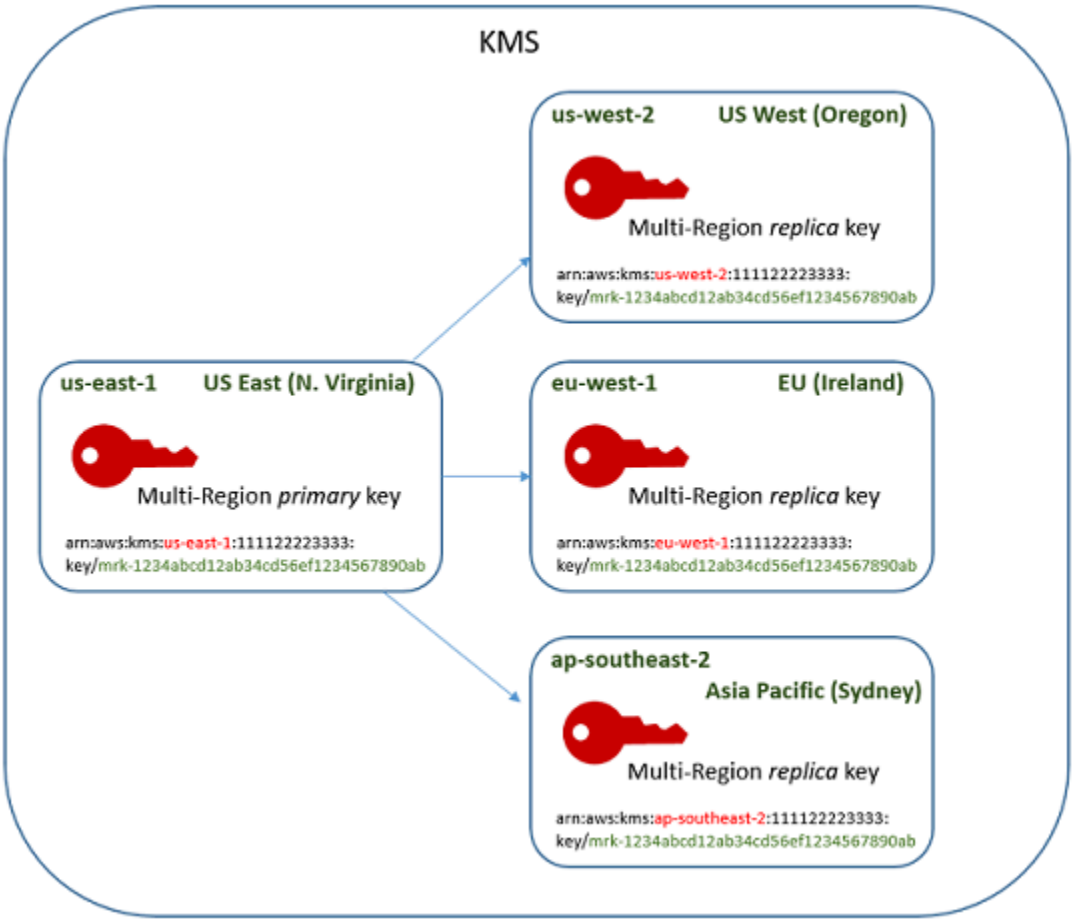
indépendante avec ses propres description, politique de clé, octrois, alias et balises. L'activation ou la désactivation d'une clé multi-région n'a aucun effet sur les clés multi-région associées. Vous pouvez utiliser les clés principales et de réplica indépendamment dans les opérations cryptographiques ou coordonner leur utilisation. Par exemple, vous pouvez chiffrer des données avec la clé principale dans la région USA Est (Virginie du Nord), déplacer les données vers la région UE (Irlande) et utiliser la clé de réplica pour déchiffrer les données.

Les clés multi-région associées ont le même ID de clé. Leurs ARN de clé (Amazon Resource Names) ne diffèrent que dans le champ Région. Par exemple, la clé principale multi-région et les clés de réplica peuvent avoir les ARN de clé suivants. L'ID de clé (le dernier élément de l'ARN de clé) est identique. Les deux clés ont l'ID de clé distinctif des clés multi-régions, qui commence par mrk-.

```
Primary key: arn:aws:kms:us-
east-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
Replica key: arn:aws:kms:eu-
west-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
```

Le même ID de clé est requis pour l'interopérabilité. Lors du chiffrement, AWS KMS lie l'ID de clé de la clé KMS au texte chiffré afin que le texte chiffré puisse être déchiffré uniquement avec cette clé KMS ou avec une clé KMS dotée du même ID de clé. Cette fonction facilite également la reconnaissance des clés multi-région associées ainsi que leur utilisation interchangeable. Par exemple, lorsque vous les utilisez dans une application, vous pouvez faire référence aux clés multi-région associées par leur ID de clé partagé. Ensuite, si nécessaire, spécifiez la région ou l'ARN pour les distinguer.

Au fur et à mesure que vos besoins en données évoluent, vous pouvez répliquer la clé principale vers d'autres Régions AWS dans la même partition, comme USA Ouest (Oregon) et Asie-Pacifique (Sydney). Le résultat consiste en quatre clés multi-région associées avec les mêmes éléments de clé et ID de clé, comme illustré dans le diagramme suivant. Vous gérez les clés indépendamment. Vous pouvez les utiliser indépendamment ou de manière coordonnée. Par exemple, vous pouvez chiffrer des données avec la clé de réplica dans la région Asie-Pacifique (Sydney), déplacer les données vers la région USA Ouest (Oregon) et les déchiffrer avec la clé de réplica dans la région USA Ouest (Oregon).



Voici d'autres considérations pour les clés multi-région.

Synchronisation des propriétés partagées — Si une propriété partagée des clés multi-région change, AWS KMS synchronise automatiquement la modification à partir de la clé principale à l'ensemble de ses clés de réplica. Vous ne pouvez pas demander ou forcer une synchronisation des propriétés partagées. AWS KMS détecte et synchronise toutes les modifications pour vous. Vous pouvez toutefois auditer la synchronisation en utilisant l'SynchronizeMultiRegionKeyévénement dans CloudTrail les journaux.

Par exemple, si vous activez la rotation automatique des clés sur une clé principale multi-région symétrique, AWS KMS copie ce paramètre sur toutes ses clés de réplica. Lorsque les éléments de clé sont soumis à une rotation, la rotation est synchronisée entre toutes les clés multi-région associées, de sorte qu'elles continuent d'avoir les mêmes éléments de clé actuels et d'accéder à toutes les versions plus anciennes des éléments de clé. Si vous créez une nouvelle clé de réplica, elle dispose des mêmes éléments de clé actuels que toutes les clés multi-région associées et de l'accès à toutes les versions précédentes des éléments de clé. Pour plus de détails, veuillez consulter Rotation de clés multi-région.

Modification de la clé principale — Chaque ensemble de clés multi-région doit avoir exactement une clé principale. La clé principale est la seule clé qui peut être répliquée. C'est également la source des propriétés partagées de ses clés de réplica. Toutefois, vous pouvez transformerr la clé principale en un réplica et transformer l'une des clés de réplica en clé principale. Vous pouvez procéder ainsi afin de supprimer une clé principale multi-région d'une région particulière ou de placer la clé principale dans une région plus proche des administrateurs de projet. Pour plus de détails, veuillez consulter Mise à jour de la région principale.

Suppression de clés multi-région — Comme toutes les clés KMS, vous devez planifier la suppression des clés multi-région avant que AWS KMS ne les supprime. Lorsque la clé est en attente de suppression, vous ne pouvez pas l'utiliser dans une opération de chiffrement. Toutefois, AWS KMS ne supprimera pas une clé principale multi-région tant que toutes ses clés de réplica n'auront pas été supprimées. Pour plus de détails, veuillez consulter Suppression de clés multi-régions.

Concepts

Les termes et concepts suivants sont utilisés avec des clés multi-région.

Clé multi-région

Une clé multi-région fait partie d'un ensemble de clés KMS avec le même ID de clé et les mêmes éléments de clé (et d'autres propriétés partagées) dans différentes Régions AWS. Chaque clé multi-région est une clé KMS pleinement fonctionnelle qui peut être utilisée indépendamment des clés multi-région associées. Puisque toutes les clés multi-région associées ont le même ID de clé et les mêmes éléments de clé, elles sont interopérables, c'est-à-dire que toute clé multi-région associée dans n'importe quelle Région AWS peut déchiffrer le texte chiffré par toute autre clé multi-région associée.

Vous définissez la propriété multi-région d'une clé KMS lors de sa création. Vous ne pouvez pas modifier propriété multi-région sur une clé existante. Vous ne pouvez pas convertir une clé à région unique en clé multi-région ou convertir une clé multi-région en clé à région unique. Pour déplacer des charges de travail existantes dans des scénarios multi-région, vous devez chiffrer à nouveau vos données ou créer de nouvelles signatures avec de nouvelles clés multi-région.

Une clé multi-région peut être symétrique ou asymétrique et peut utiliser des éléments de clé AWS KMSou des éléments de clé importés. Vous ne pouvez pas créer de clés multi-région dans un magasin de clés personnalisé.

Dans un ensemble de clés multi-région associées, il y a exactement une clé principale à tout moment. Vous pouvez créer des clés de réplica de cette clé principale dans d'autres Régions AWS.

Vous pouvez également mettre à jour la région principale, qui transforme la clé principale en clé de réplica et transforme une clé de réplica spécifiée en clé principale. Toutefois, vous ne pouvez conserver qu'une seule clé principale ou clé de réplica dans chaque Région AWS. Toutes les régions doivent être dans la même partition AWS.

Vous pouvez avoir plusieurs ensembles de clés multi-région associées dans la même ou dans plusieurs Régions AWS. Bien que les clés multi-région associées soient interopérables, les clés multi-région non associées ne sont pas interopérables.

Clé primaire

Une clé principale multi-région est une clé KMS qui peut être répliquée dans d'autres Régions AWS de la même partition. Chaque ensemble de clés multi-région ne possède qu'une seule clé principale.

Une clé principale diffère d'une clé de réplica comme suit :

- Seule une clé principale peut être répliquée.
- La clé principale est la source de propriétés partagées de ses clés de réplica, y compris les éléments de clé et l'ID de clé.
- Vous pouvez activer et désactiver la rotation automatique des clés seulement sur une clé principale.
- Vous pouvez planifier la suppression d'une clé principale à tout moment. Mais AWS KMS ne supprimera pas une clé primaire tant que toutes ses clés de réplica n'auront pas été supprimées.

Cependant, les clés principales et les clés de réplica ne diffèrent pas au niveau des propriétés cryptographiques. Vous pouvez utiliser une clé primaire et ses clés de réplica de manière interchangeable.

Vous n'êtes pas tenu de répliquer une clé principale. Vous pouvez l'utiliser comme n'importe quelle clé KMS et la répliquer si elle est utile. Toutefois, étant donné que les clés multi-région ont des propriétés de sécurité différentes de celles des clés à région unique, nous vous recommandons de créer une clé multi-région uniquement lorsque vous envisagez de la répliquer.

Clé de réplica

Une clé de réplica multi-région est une clé KMS qui a les mêmes ID de clé et éléments de clé que sa clé principale et les clés de réplica associées, mais qui existe dans une Région AWS différente.

Une clé de réplica est une clé KMS pleinement fonctionnelle avec ses propres politiques de clé, octrois, alias, balises et autres propriétés. Il ne s'agit pas d'une copie ou d'un pointeur vers la clé principale ou toute autre clé. Vous pouvez utiliser une clé de réplica même si sa clé principale et toutes les clés de réplica associées sont désactivées. Vous pouvez également transformer une clé de réplica en clé principale et une clé principale en clé de réplica. Une fois créée, une clé de réplica s'appuie sur sa clé principale uniquement pour la rotation des clés et la mise à jour de la région principale.

Les clés principales et les clés de réplica ne diffèrent pas au niveau des propriétés cryptographiques. Vous pouvez utiliser une clé primaire et ses clés de réplica de manière interchangeable. Les données chiffrées par une clé principale ou de réplica peuvent être déchiffrées par la même clé, ou par toute clé principale ou de réplica associée.

Répliquer

Vous pouvez répliquer une clé principale multi-région dans une autre Région AWS de la même partition. Dans ce cas, AWS KMS crée une clé de réplica multi-région dans la région spécifiée, avec le même ID de clé et d'autres propriétés partagées en tant que clé primaire. Ensuite, il transporte en toute sécurité les éléments de clé au-delà des limites de la région et les associe à la nouvelle clé de réplica, le tout dans AWS KMS.

Propriétés partagées

Les propriétés partagées sont des propriétés d'une clé principale multi-région qui sont partagées avec ses clés de réplica. AWS KMS crée les clés de réplica avec les mêmes valeurs de propriété partagées que celles de la clé principale. Ensuite, il synchronise périodiquement les valeurs de propriété partagées de la clé principale avec ses clés de réplica. Vous ne pouvez pas définir ces propriétés sur une clé de réplica.

Voici les propriétés partagées des clés multi-région.

- ID de clé — (L'élément `Region` de l'ARN de clé diffère.)
- Éléments de clé
- Origine des éléments de clé
- Spécifications des clés et algorithmes de chiffrement
- Utilisation de la clé
- Rotation automatique des clés — Vous pouvez activer et désactiver la rotation automatique des clés uniquement sur la clé principale. De nouvelles clés de réplica sont créées avec toutes les

versions des éléments de clé partagés. Pour plus de détails, veuillez consulter Rotation de clés multi-région.

Vous pouvez également considérer les désignations principales et de réplica des clés multi-région associées comme des propriétés partagées. Lorsque vous créez de nouvelles clés de réplica ou mettez à jour la clé principale, AWS KMSsynchronise la modification sur toutes les clés multi-région associées. Lorsque ces modifications sont terminées, toutes les clés multi-région associées répertorient avec précision leur clé principale et leurs clés de réplica.

Toutes les autres propriétés des clés multi-région sont des propriétés indépendantes, y compris la description, la politique de clé, les octrois, les états de clé activé et désactivé, les alias et les balises. Vous pouvez définir les mêmes valeurs pour ces propriétés sur toutes les clés multi-région associées, mais si vous modifiez la valeur d'une propriété indépendante, AWS KMS ne la synchronise pas.

Vous pouvez suivre la synchronisation des propriétés partagées de vos clés multi-région. Dans votre AWS CloudTrail journal, recherchez l'SynchronizeMultiRegionKeyévénement.

Contrôle de l'accès aux clés multi-régions

Vous pouvez utiliser des clés multi-région dans des scénarios de conformité, de reprise après sinistre et de sauvegarde qui seraient plus complexes avec les clés à région unique. Toutefois, étant donné que les propriétés de sécurité des clés multi-région sont significativement différentes de celles des clés à région unique, nous vous recommandons de faire preuve de prudence lorsque vous autorisez la création, la gestion et l'utilisation de clés multi-région.

> **Note**
>
> Les instructions de politique IAM existantes avec des caractères génériques dans le champ `Resource` s'appliquent désormais à la fois aux clés à région unique et multi-région. Pour les limiter aux clés KMS à région unique ou aux clés multirégionales, utilisez la clé de MultiRegion condition kms :.

Utilisez vos outils d'autorisation pour empêcher la création et l'utilisation de clés multi-région dans tous les scénarios où une clé à région unique suffira. Autoriser les principaux à répliquer une clé multi-région uniquement dans les Régions AWS qui l'exigent. Donnez l'autorisation pour les clés multi-région uniquement aux principaux qui en ont besoin et uniquement pour les tâches qui en ont besoin.

Vous pouvez utiliser des politiques de clé, des politiques IAM et des octrois pour permettre aux principaux IAM de gérer et d'utiliser des clés multi-région dans votre Compte AWS. Chaque clé multi-région est une ressource indépendante dotée d'un ARN de clé unique et d'une politique de clé. Vous devez établir et maintenir une stratégie de clé pour chaque clé et vous assurer que les stratégies IAM nouvelles et existantes mettent en œuvre votre stratégie d'autorisation.

Rubriques

- Notions de base sur les autorisations pour les clés multi-région
- Autorisation des administrateurs et des utilisateurs de clés multi-région
- Autoriser AWS KMS à synchroniser de clés multi-région

Notions de base sur les autorisations pour les clés multi-région

Lors de la conception de politiques de clé et de politiques IAM pour les clés multi-région, tenez compte des principes suivants.

- Politique de clé — Chaque clé multi-région est une ressource de clé KMS indépendante avec sa propre politique de clé. Vous pouvez appliquer la même politique de clé ou une politique de clé différente à chaque clé de l'ensemble des clés multi-région associées. Les politiques de clé ne sont pas des propriétés partagées des clés multi-région. AWS KMS ne copie ni ne synchronise les politiques de clé entre les clés multi-région associées.

  Lorsque vous créez une clé de réplica dans la console AWS KMS, celle-ci affiche la politique de clé actuelle de la clé principale à titre de commodité. Vous pouvez utiliser cette politique de clé, la modifier ou la supprimer et la remplacer. Mais même si vous acceptez la politique de clé principale inchangée, AWS KMS ne synchronise pas les politiques. Par exemple, si vous modifiez la politique de clé de la clé principale, la politique de clé de la clé de réplica reste la même.
- Politique clé par défaut — Lorsque vous créez des clés multirégionales à l'aide `ReplicateKey` des opérations CreateKeyet, la politique clé par défaut est appliquée sauf si vous spécifiez une stratégie clé dans la demande. Il s'agit de la même politique de clé par défaut qui est appliquée aux clés à région unique.
- Politiques IAM — Comme pour toutes les clés KMS, vous pouvez utiliser des politiques IAM pour contrôler l'accès aux clés multi-région uniquement lorsque la politique de clé le permet. Les politiques IAM s'appliquent à l'ensemble des Régions AWS par défaut. Cependant, vous pouvez utiliser des clés de condition, telles que aws : RequestedRegion, pour limiter les autorisations à une région particulière.

Pour créer des clés principales et des clés de réplica, les principaux doivent avoir l'autorisation `kms:CreateKey` dans une politique IAM qui s'applique à la région où la clé est créée.

- Octrois — Les octrois AWS KMS sont régionaux. Chaque octroi autorise l'ajout d'autorisations sur une clé KMS. Vous pouvez utiliser des octrois pour autoriser des autorisations sur une clé principale ou une clé de réplica multi-région. Mais vous ne pouvez pas utiliser un seul octroi pour autoriser des autorisations sur plusieurs clés KMS, même s'il s'agit de clés multi-région associées.
- ARN de clé — Chaque clé multi-région a un ARN de clé unique. Les ARN de clé des clés multi-région associées ont les mêmes partition, compte et ID de clé, mais des régions différentes.

  Pour appliquer une instruction de politique IAM à une clé multi-région particulière, utilisez son ARN de clé ou un modèle d'ARN de clé qui inclut la région. Pour appliquer une instruction de politique IAM à toutes les clés multi-région associées, utilisez un caractère générique (*) dans l'élément Region de l'ARN, comme illustré dans l'exemple suivant.

  ```
  {
    "Effect": "Allow",
    "Action": [
      "kms:Describe*",
      "kms:List*"
    ],
    "Resource": {
        "arn:aws:kms:*::111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
    }
  }
  ```

  Pour appliquer une déclaration de politique à toutes les clés multirégionales de votre cléCompte AWS, vous pouvez utiliser la condition de MultiRegion politique kms : ou un modèle d'identification de clé incluant le `mrk-` préfixe distinctif.
- Rôle lié à un service — Les principaux qui créent des clés primaires multirégionales doivent disposer de l'autorisation iam :. CreateServiceLinkedRole

  Pour synchroniser les propriétés partagées des clés multi-région associées, AWS KMS endosse un rôle lié à un service IAM. AWS KMS crée le rôle lié à un service dans le Compte AWS chaque fois que vous créez une clé principale multi-région. (Si le rôle existe, AWS KMS le recrée, ce qui n'a aucun effet nocif.) Le rôle est valable dans toutes les régions. Pour permettre AWS KMS de créer (ou de recréer) le rôle lié au service, les principaux qui créent des clés primaires multirégionales doivent disposer de l'autorisation iam :. CreateServiceLinkedRole

Autorisation des administrateurs et des utilisateurs de clés multi-région

Les principaux qui créent et gèrent des clés multi-région ont besoin des autorisations suivantes dans les régions principale et de réplica :

- `kms:CreateKey`
- `kms:ReplicateKey`
- `kms:UpdatePrimaryRegion`
- `iam:CreateServiceLinkedRole`

Création d'une clé principale

Pour créer une clé primaire multirégionale, le principal a besoin des CreateServiceLinkedRole autorisations kms : CreateKey et iam : dans le cadre d'une politique IAM effective dans la région de la clé primaire. Les principaux qui disposent de ces autorisations peuvent créer des clés à région unique et multi-région à moins que vous ne restreigniez leurs autorisations.

L'`iam:CreateServiceLinkedRole`autorisation permet AWS KMS de créer le AWSServiceRoleForKeyManagementServiceMultiRegionKeysrôle pour synchroniser les propriétés partagées des clés multirégionales associées.

Par exemple, cette politique IAM permet à un principal de créer n'importe quel type de clé KMS.

```
{
  "Version": "2012-10-17",
  "Statement":{
      "Action": [
        "kms:CreateKey",
        "iam:CreateServiceLinkedRole"
      ],
      "Effect":"Allow",
      "Resource":"*"
  }
}
```

Pour autoriser ou refuser l'autorisation de créer des clés primaires multirégionales, utilisez la clé de MultiRegion condition kms :. Les valeurs valides sont `true` (clé multi-région) ou `false` (clé à région unique). Par exemple, l'instruction de politique IAM utilise une action `Deny` avec la clé de condition `kms:MultiRegion` pour empêcher les principaux de créer des clés multi-région.

```
{
  "Version": "2012-10-17",
  "Statement":{
      "Action":"kms:CreateKey",
      "Effect":"Deny",
      "Resource":"*",
      "Condition": {
          "Bool": "kms:MultiRegion": true
      }
  }
}
```

Réplication de clés

Pour créer une clé de réplica multi-région, le principal a besoin des autorisations suivantes :

- kms : ReplicateKey autorisation dans la politique de clé de la clé primaire.
- kms : CreateKey autorisation dans une politique IAM en vigueur dans la région de la clé de réplique.

Soyez prudent lorsque vous autorisez ces autorisations. Elles permettent aux principaux de créer des clés KMS et les politiques de clé qui autorisent leur utilisation. L'autorisation `kms:ReplicateKey` permet également le transfert d'éléments de clé au-delà des limites de la région dans AWS KMS.

Pour limiter les limites Régions AWS dans lesquelles une clé multirégionale peut être répliquée, utilisez la clé de ReplicaRegion condition kms :. Elle ne limite que l'autorisation `kms:ReplicateKey`. Sinon, elle n'a aucun effet. Par exemple, la politique de clé suivante autorise le principal à répliquer cette clé principale, mais uniquement dans les régions spécifiées.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
         "us-east-1",
```

```
            "eu-west-3",
            "ap-southeast-2"
        ]
      }
    }
}
```

Mise à jour de la région principale

Les principaux autorisés peuvent transformer une clé de réplica en clé principale, ce qui transforme l'ancienne clé principale en un réplica. Cette action s'appelle la mise à jour de la région principale. Pour mettre à jour la région principale, le principal a besoin d'une UpdatePrimaryRegion autorisation de km : dans les deux régions. Vous pouvez fournir ces autorisations dans une politique de clé ou une politique IAM.

- `kms:UpdatePrimaryRegion` sur la clé principale. Cette autorisation doit être effective dans la région de clé principale.
- `kms:UpdatePrimaryRegion` sur la clé de réplica. Cette autorisation doit être effective dans la région de clé de réplica.

Par exemple, la politique de clé suivante donne aux utilisateurs qui peuvent endosser le rôle Administrateur l'autorisation de mettre à jour la région principale de la clé KMS. Cette clé KMS peut être la clé principale ou une clé de réplica dans cette opération.

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}
```

Pour restreindre la Régions AWS capacité d'héberger une clé primaire, utilisez la clé de PrimaryRegion condition kms :. Par exemple, l'instruction de politique IAM suivante permet principaux de mettre à jour la région principale des clés multi-région dans le Compte AWS, mais uniquement lorsque la nouvelle région principale est l'une des régions spécifiées.

```
{
```

```
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Resource": {
      "arn:aws:kms:*:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-west-2",
         "sa-east-1",
         "ap-southeast-1"
      ]
    }
  }
}
```

Utilisation et gestion des clés multi-région

Par défaut, les principaux qui ont l'autorisation d'utiliser et de gérer les clés KMS dans un Compte AWS et une région ont également l'autorisation d'utiliser et de gérer des clés multi-région. Cependant, vous pouvez utiliser la clé de MultiRegion condition kms : pour autoriser uniquement les clés à région unique ou uniquement les clés multirégionales. Vous pouvez également utiliser la clé de MultiRegionKeyType condition kms : pour autoriser uniquement les clés primaires multirégionales ou uniquement les clés de réplique. Les deux clés de condition contrôlent l'accès à l'CreateKeyopération et à toute opération utilisant une clé KMS existante, telle que Encrypt ou EnableKey.

L'exemple suivant d'instruction de politique IAM utilise la clé de condition `kms:MultiRegion` pour empêcher les principaux d'utiliser ou de gérer une clé multi-région.

```
{
  "Effect": "Deny",
  "Action": "kms:*",
  "Resource": "*",
  "Condition": {
    "Bool": "kms:MultiRegion": true
  }
}
```

Cet exemple d'instruction de politique IAM utilise la condition `kms:MultiRegionKeyType` pour permettre aux principaux de planifier et d'annuler la suppression de clé, mais uniquement sur les clés de réplica multi-région.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
      "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Autoriser AWS KMS à synchroniser de clés multi-région

Afin de prendre en charge les clés multi-région, AWS KMS utilise un rôle lié à un service IAM. Ce rôle donne à AWS KMS les autorisations dont il a besoin pour synchroniser les propriétés partagées. Vous pouvez consulter l'SynchronizeMultiRegionKey CloudTrail événement qui enregistre la AWS KMS synchronisation des propriétés partagées dans vos AWS CloudTrail journaux.

À propos du rôle lié à un service pour les clés multi-région

Un rôle lié à un service est un rôle IAM qui accorde à un service AWS l'autorisation d'appeler d'autres services AWS en votre nom. Il est conçu pour faciliter l'utilisation des fonctions de plusieurs services AWS intégrés, sans avoir à créer et gérer des politiques IAM complexes.

Pour les clés multirégionales, AWS KMS crée le rôle AWSServiceRoleForKeyManagementServiceMultiRegionKeyslié au service avec la politique. AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy Cette politique donne au rôle l'autorisation `kms:SynchronizeMultiRegionKey`, qui lui permet de synchroniser les propriétés partagées des clés multi-région.

Étant donné que le rôle AWSServiceRoleForKeyManagementServiceMultiRegionKeyslié au service n'est fiable que`mrk.kms.amazonaws.com`, seul le rôle lié au service AWS KMS peut assumer ce rôle lié au service. Ce rôle est limité aux opérations dont AWS KMS a besoin pour synchroniser les propriétés partagées multi-région. Aucune autre autorisation n'est accordée à AWS KMS. Par exemple, AWS KMS n'a pas l'autorisation de créer, répliquer ou supprimer des clés KMS.

Pour de plus amples informations sur l'utilisation des rôles liés à un service par les services AWS, veuillez consulter Utilisation des rôles liés à un service dans le Guide de l'utilisateur IAM.

Création du rôle lié à un service

AWS KMScrée automatiquement le rôle AWSServiceRoleForKeyManagementServiceMultiRegionKeyslié au service dans votre Compte AWS lorsque vous créez une clé multirégionale, si le rôle n'existe pas déjà. Vous ne pouvez pas créer ou recréer directement ce rôle lié à un service.

Modifier la description du rôle lié à un service

Vous ne pouvez pas modifier le nom du rôle ni les déclarations de politique du rôle AWSServiceRoleForKeyManagementServiceMultiRegionKeyslié à un service, mais vous pouvez modifier la description du rôle. Pour de plus amples informations, veuillez consulter Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Supprimer le rôle lié à un service

AWS KMSne supprime pas le rôle AWSServiceRoleForKeyManagementServiceMultiRegionKeyslié au service de votre compte Compte AWS et vous ne pouvez pas le supprimer. Cependant, AWS KMS n'assume pas le AWSServiceRoleForKeyManagementServiceMultiRegionKeysrôle et n'utilise aucune de ses autorisations, sauf si vous avez des clés multirégionales dans votre région Compte AWS et.

Création de clés multi-régions

Vous pouvez créer des clés multi-région dans la console ou via l'API AWS KMS.

La propriété multi-région que vous définissez dans cette procédure est inaltérable. Vous ne pouvez pas convertir une clé à région unique en clé multi-région ou convertir une clé multi-région en clé à région unique.

Rubriques

- Création de clés principales multi-région
- Création de clés de réplica multi-région

Création de clés principales multi-région

Vous pouvez créer une clé principale multi-région dans la console AWS KMS ou à l'aide de l'API AWS KMS. Vous pouvez créer la clé principale dans n'importe quelle Région AWS où AWS KMS prend en charge les clés multi-région.

Pour créer une clé primaire multirégionale, le principal a besoin des mêmes autorisations que celles dont il a besoin pour créer n'importe quelle clé KMS, y compris l'CreateKeyautorisation kms : dans une politique IAM. Le principal a également besoin de l'CreateServiceLinkedRoleautorisation iam :. Vous pouvez utiliser la clé de MultiRegionKeyType condition kms : pour autoriser ou refuser l'autorisation de créer des clés primaires multirégionales.

Ces instructions créent une clé principale multi-région avec des éléments de clé générés par AWS KMS. Pour créer une clé principale multi-région avec des éléments de clé importés, veuillez consulter Création d'une clé principale avec des éléments de clé importés.

Rubriques

- Création d'une clé principale multi-région (console)
- Création d'une clé principale multi-région (API AWS KMS)

Création d'une clé principale multi-région (console)

Pour créer une clé principale multi-région dans la console AWS KMS, utilisez le même processus que vous utiliseriez pour créer n'importe quelle clé KMS. Vous sélectionnez une clé multi-région dans Advanced options (Options avancées). Pour obtenir des instructions complètes, veuillez consulter Création de clés.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key.
5. Sélectionnez un type de clé symétrique ou asymétrique. Les clés symétriques sont les valeurs par défaut.

Vous pouvez créer des clés symétriques et asymétriques multi-région, y compris des clés HMAC KMS multi-région, qui sont symétriques.

6. Sélectionnez l'utilisation de vos clés. Encrypt and decrypt (Chiffrer et déchiffrer) est la valeur par défaut.

   Pour obtenir de l'aide, veuillez consulter the section called “Création de clés”, the section called “Création de clés KMS asymétriques” ou the section called “Création de clés HMAC”.

7. (Facultatif) Développez Options avancées.

8. Pour que AWS KMS génère les éléments de clé que vos clés principales et de réplica partageront, sélectionnez KMS sous Key material origin (Origine des éléments de clé). Si vous importez des éléments de clé dans les clés principales et clés de réplica, sélectionnez External (Import key material) (Externe (Importation des éléments de clé)).

9. Sous Multi-Region replication (Réplication multi-région), choisissez Allow this key to be replicated into other Regions (Autoriser cette clé à être répliquée dans d'autres régions).

   Vous ne pouvez pas modifier ce paramètre une fois que vous créez la clé KMS.

10. Saisissez un alias pour la clé principale.

    Les alias ne sont pas une propriété partagée de clés multi-région. Vous pouvez attribuer à votre clé principale multi-région et à ses réplicas le même alias ou des alias différents. AWS KMS ne synchronise pas les alias des clés multi-région.

    > **Note**
    >
    > L'ajout, la suppression ou la mise à jour d'un alias peut permettre d'accorder ou de refuser l'autorisation d'utiliser la KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation d'alias pour contrôler l'accès aux clés KMS.

11. (Facultatif) Saisissez une description de la clé principale.

    Les descriptions ne sont pas une propriété partagée des clés multi-région. Vous pouvez donner à votre clé principale multi-région et à ses réplicas la même description ou des descriptions différentes. AWS KMS ne synchronise pas les descriptions de clé des clés multi-région.

12. (Facultatif) Saisissez une clé de balise et une valeur de balise facultative. Pour affecter plus d'une balise à la clé principale, sélectionnez Add tag (Ajouter une balise).

Les balises ne sont pas une propriété partagée des clés multi-région. Vous pouvez attribuer à votre clé principale multi-région et à ses réplicas les mêmes balises ou des balises différentes. AWS KMS ne synchronise pas les balises des clés multi-région. Vous pouvez modifier les balises des clés KMS à tout moment.

> **Note**
>
> L'ajout ou la suppression d'une balise sur une KMS permet d'accorder ou de refuser l'autorisation d'utilisation de cette clé KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation de balises pour contrôler l'accès aux clés KMS.

13. Sélectionnez les utilisateurs et les rôles IAM qui peuvent administrer la clé principale.

> **Note**
>
> Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles l'autorisation de gérer la clé KMS.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

Cette étape démarre le processus de création d'une politique de clé pour la clé principale. Les politiques de clé ne sont pas une propriété partagée des clés multi-région. Vous pouvez attribuer à votre clé principale multi-région et à ses réplicas la même politique de clé ou des politiques de clé différentes. AWS KMS ne synchronise pas les politiques de clé des clés multi-région. Vous pouvez modifier la politique de clé d'une clé KMS à tout moment.

14. Suivez les étapes de création de la politique de clé, y compris sur la sélection des utilisateurs de clés. Après avoir passé en revue la politique de clé, choisissez Finish (Terminer) pour créer la clé KMS.

Création d'une clé principale multi-région (API AWS KMS)

Pour créer une clé primaire multirégionale, utilisez l'CreateKeyopération. Utilisez le paramètre `MultiRegion` avec la valeur `True`.

Par exemple, la commande suivante crée une clé principale multi-région dans la Région AWS (us-east-1) de l'appelant. Elle accepte les valeurs par défaut pour toutes les autres propriétés, y compris la politique de clé. Les valeurs par défaut pour les clés principales multi-région sont les mêmes que les valeurs par défaut pour toutes les autres clés KMS, y compris la politique de clé par défaut. Cette procédure crée une clé de chiffrement symétrique, la clé KMS par défaut.

La réponse inclut l'élément `MultiRegion` et l'élément `MultiRegionConfiguration` avec des sous-éléments et des valeurs typiques pour une clé principale multi-région sans clés de réplica. L'ID de clé d'une clé multi-région commence toujours par `mrk-`.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans les champs `Description` ou `Tags`. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

```
$ aws kms create-key --multi-region
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1606329032.475,
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
```

```
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [ ]
      }
    }
}
```

Création de clés de réplica multi-région

Vous pouvez créer une réplique de clé multirégionale dans la AWS KMS console, en utilisant l'ReplicateKeyopération ou en utilisant un AWS CloudFormationmodèle. Vous ne pouvez pas utiliser CreateKeycette opération pour créer une clé de réplique.

Vous pouvez utiliser ces procédures pour répliquer n'importe quelle clé principale multi-région, y compris une clé KMS de chiffrement symétrique, une clé KMS asymétrique ou une clé KMS HMAC.

Lorsque cette opération est terminée, la nouvelle clé de réplica a un état de clé de `Creating`. Cet état de clé passe à `Enabled` (ou PendingImport) au bout de quelques secondes lorsque le processus de création de la nouvelle clé de réplique est terminé. Alors que l'état de la clé est `Creating`, vous pouvez gérer la clé, mais vous ne pouvez pas encore l'utiliser dans les opérations cryptographiques. Si vous créez et utilisez la clé de réplique par programmation, réessayez `KMSInvalidStateException` ou appelez DescribeKeypour vérifier sa `KeyState` valeur avant de l'utiliser.

Si vous supprimez par erreur une clé de réplica, vous pouvez utiliser cette procédure pour la recréer. Si vous répliquez la même clé primaire dans la même région, la nouvelle clé de réplica que vous allez créer aura les mêmes propriétés partagées que la clé de réplica d'origine.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

En savoir plus

- Pour créer une clé de réplica multi-région avec des éléments de clé importés, veuillez consulter Création d'une clé de réplica avec des éléments de clé importés.
- Pour utiliser un AWS CloudFormation modèle afin de créer une réplique de clé, reportez-vous AWS::KMS::ReplicaKeyau guide de l'AWS CloudFormationutilisateur.

Rubriques

- Régions de réplica
- Création de clés de réplica (console)
- Création d'une clé de réplica (API AWS KMS)

Régions de réplica

Vous choisissez généralement de répliquer une clé multi-région dans une Région AWS en fonction de votre modèle économique et de vos exigences réglementaires. Par exemple, vous pouvez répliquer une clé dans les régions où vous conservez vos ressources. Ou, pour vous conformer à une exigence de reprise après sinistre, vous pouvez répliquer une clé dans des régions géographiquement éloignées.

Les éléments suivants sont les exigences AWS KMS pour les régions de réplica. Si la région que vous choisissez n'est pas conforme à ces exigences, les tentatives de réplication d'une clé échouent.

- Une clé multi-région associée par région — Vous ne pouvez pas créer de clé de réplica dans la même région que sa clé principale ou dans la même région qu'un autre réplica de la clé principale.

  Si vous essayez de répliquer une clé primaire dans une région qui possède déjà un réplica de cette clé, la tentative échouera. Si la clé de réplica actuelle dans la région affiche l'état de clé `PendingDeletion`, vous pouvez annuler la suppression de la clé de réplica ou attendre que la clé de réplica soit supprimée.
- Plusieurs clés multi-région non associées dans la même région — Vous pouvez avoir plusieurs clés multi-région non associées dans la même région. Vous pouvez par exemple avoir deux clés principales multi-région dans la région `us-east-1`. Chacune des clés principales peut avoir une clé de réplica dans la région `us-west-2`.
- Régions dans la même partition — La région de clé de réplica doit être dans la même partition AWS que la région de clé principale.
- La région doit être activée — Si une région est désactivée par défaut, vous ne pouvez pas créer de ressources dans cette région tant qu'elle n'est pas activée pour votre Compte AWS.

Création de clés de réplica (console)

Dans la console AWS KMS, vous pouvez créer un ou plusieurs réplicas d'une clé principale multi-région dans la même opération.

Cette procédure est similaire à la création d'une clé KMS standard à région unique dans la console. Toutefois, comme une clé de réplica est basée sur la clé principale, vous ne sélectionnez pas de valeurs pour les propriétés partagées, telles que les spécifications (symétrique ou asymétrique), l'utilisation ou l'origine de la clé.

Vous spécifiez des propriétés qui ne sont pas partagées, notamment un alias, des balises, une description et une politique de clé. Par commodité, la console affiche les valeurs de propriété actuelles de la clé principale, mais vous pouvez les modifier. Même si vous conservez les valeurs de clé principale, AWS KMS ne conserve pas ces valeurs synchronisées.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Sélectionnez l'ID de clé ou l'alias d'une clé principale multi-région. Cela ouvre la page des détails de clé pour la clé KMS.

   Pour identifier une clé principale multi-région, utilisez l'icône de l'outil dans le coin supérieur droit pour ajouter la colonne Regionality (Régionalité) dans la table.
5. Cliquez sur l'onglet Regionality (Régionalité).
6. Dans la section Related multi-Region keys (Clés multi-région associées), choisissez Create new replica keys (Créer de nouvelles clés de réplica).

La section Related multi-Region keys (Clés multi-région associées) affiche la région de la clé principale et de ses clés de réplica. Vous pouvez utiliser cet affichage pour vous aider à choisir la région pour votre nouvelle clé de réplica.

7. Sélectionnez une ou plusieurs Régions AWS. Cette procédure crée une clé de réplica dans chacune des régions que vous sélectionnez.

   Le menu inclut uniquement les régions dans la même partition AWS que la clé principale. Les régions qui ont déjà une clé multi-région associée sont affichées, mais ne peuvent pas être sélectionnées. Vous n'êtes peut-être pas autorisé à répliquer une clé dans toutes les régions du menu.

   Lorsque vous avez terminé de choisir les régions, fermez le menu. Les régions que vous avez choisies s'affichent. Pour annuler la réplication dans une région, cliquez sur le X en regard du nom de la région.

8. Saisissez un alias pour la clé de réplica.

   La console affiche l'un des alias actuels de la clé principale, mais vous pouvez le modifier. Vous pouvez attribuer à votre clé principale multi-région et à ses réplicas le même alias ou des alias différents. Les alias ne sont pas une propriété partagée des clés multi-région. AWS KMS ne synchronise pas les alias des clés multi-région.

   L'ajout, la suppression ou la mise à jour d'un alias peut permettre d'accorder ou de refuser l'autorisation d'utiliser la KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation d'alias pour contrôler l'accès aux clés KMS.

9. (Facultatif) Saisissez une description de la clé de réplica.

   La console affiche la description actuelle de la clé principale, mais vous pouvez la modifier. Les descriptions ne sont pas une propriété partagée des clés multi-région. Vous pouvez donner à votre clé principale multi-région et à ses réplicas la même description ou des descriptions différentes. AWS KMS ne synchronise pas les descriptions de clé des clés multi-région.

10. (Facultatif) Saisissez une clé de balise et une valeur de balise facultative. Pour affecter plus d'une balise à la clé de réplica, sélectionnez Add tag (Ajouter une balise).

    La console affiche les balises actuellement attachées à la clé principale, mais vous pouvez les modifier. Les balises ne sont pas une propriété partagée des clés multi-région. Vous pouvez attribuer à votre clé principale multi-région et à ses réplicas les mêmes balises ou des balises différentes. AWS KMS ne synchronise pas les balises des clés multi-région.

L'étiquetage ou le désétiquetage d'une clé KMS permet d'accorder ou refuser l'autorisation d'utilisation de cette clé KMS. Pour plus de détails, veuillez consulter ABAC pour AWS KMS et Utilisation de balises pour contrôler l'accès aux clés KMS.

11. Sélectionnez les utilisateurs et les rôles IAM qui peuvent administrer la clé de réplica.

> **Note**
>
> Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation de gérer les clés de réplica.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

Cette étape démarre le processus de création d'une politique de clé pour la clé de réplica. La console affiche la politique de clé actuelle de la clé principale, mais vous pouvez la modifier. Les politiques de clé ne sont pas une propriété partagée des clés multi-région. Vous pouvez attribuer à votre clé principale multi-région et à ses réplicas la même politique de clé ou des politiques de clé différentes. AWS KMS ne synchronise pas les politiques de clé. Vous pouvez modifier la politique de clé d'une clé KMS à tout moment.

12. Suivez les étapes de création de la politique de clé, y compris sur la sélection des utilisateurs de clés. Après avoir passé en revue la politique de clé, choisissez Finish (Terminer) pour créer la clé de réplica.

Création d'une clé de réplica (API AWS KMS)

Pour créer une clé de réplique multirégionale, utilisez l'ReplicateKeyopération. Vous ne pouvez pas utiliser CreateKeycette opération pour créer une clé de réplique. Cette opération crée une clé de réplica à la fois. La région que vous spécifiez doit respecter les exigences de région pour les clés de réplica.

Lorsque vous utilisez l'opération `ReplicateKey`, vous ne spécifiez aucune valeur pour les propriétés partagées des clés multi-région. Les valeurs des propriétés partagées sont copiées à partir de la clé principale et leur synchronisation est maintenue. Toutefois, vous pouvez spécifier des

valeurs pour les propriétés qui ne sont pas partagées. Sinon, AWS KMS applique les valeurs par défaut standard pour les clés KMS, et non les valeurs de la clé principale.

> **Note**
>
> Si vous ne spécifiez pas de valeurs pour les paramètres `Description`, `KeyPolicy` ou `Tags`, AWS KMS crée la clé de réplica sans balises, une description de chaîne vide et la politique de clé par défaut.
> N'incluez pas d'informations confidentielles ou sensibles dans les champs `Description` ou `Tags`. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

Par exemple, la commande suivante crée une clé de réplica multi-région dans la région Asie-Pacifique (Sydney) (ap-southeast-2). Cette clé de réplica est modélisée sur la clé principale de la région USA Est (Virginie du Nord) (us-east-1), qui est identifiée par la valeur du paramètre `KeyId`. Cet exemple accepte les valeurs par défaut pour toutes les autres propriétés, y compris la politique de clé.

La réponse décrit la nouvelle clé de réplica. Elle inclut des champs pour les propriétés partagées, tels que `KeyId`, `KeySpec`, `KeyUsage`, et l'origine des éléments de clé (`Origin`). Elle inclut également des propriétés indépendantes de la clé principale, telles que la `Description`, la politique de clé (`ReplicaKeyPolicy`), et les balises (`ReplicaTags`).

La réponse inclut également l'ARN de clé et la région de la clé principale et toutes ses clés de réplica, y compris celle qui vient d'être créée dans la région ap-southeast-2. Dans cet exemple, l'élément `ReplicaKey` montre que cette clé principale a déjà été répliquée dans la région UE (Irlande) (eu-west-1).

```
$ aws kms replicate-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
    --replica-region ap-southeast-2
{
    "ReplicaKeyMetadata": {
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "REPLICA",
            "PrimaryKey": {
```

```
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                },
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                }
            ]
        },
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1607472987.918,
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    },
    "ReplicaKeyPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-
default-1\",...,
    "ReplicaTags": []
}
```

Affichage des clés multi-régions

Vous pouvez afficher les clés à région unique et multi-région dans la console AWS KMS et avec les opérations d'API AWS KMS.

Rubriques

- Affichage des clés multi-région dans la console
- Affichage des clés multi-région dans l'API

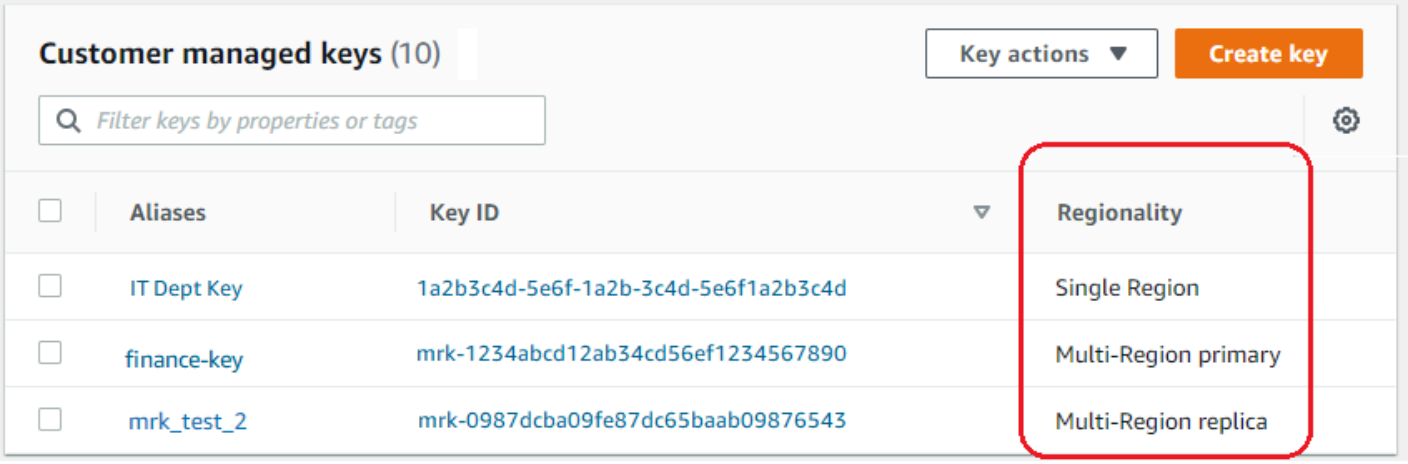
Affichage des clés multi-région dans la console

Dans la console AWS KMS, vous pouvez afficher les clés KMS dans la région sélectionnée. Toutefois, si vous avez une clé multi-région, vous pouvez voir ses clés multi-région associées dans d'autres Régions AWS.

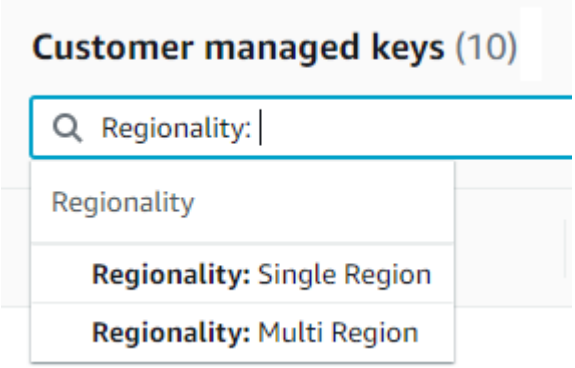
Le tableau Clés gérées par le client dans la console AWS KMS affiche uniquement les clés KMS de la région sélectionnée. Vous pouvez afficher les clés principales et de réplica multi-région dans la région sélectionnée. Pour changer de Région AWS, utilisez le sélecteur de Région dans l'angle supérieur droit de la page.

Le tableau Clés gérées par AWS n'a pas les fonctions de régionalité, car les Clés gérées par AWS sont toujours des clés à région unique.

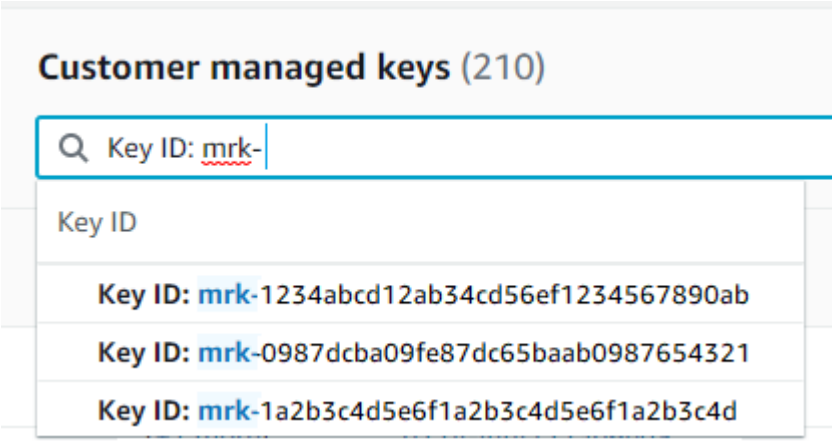
- Pour faciliter l'identification de vos clés multi-région, ajoutez la colonne Regionality (Régionalité) à votre table de clés. Pour obtenir de l'aide, veuillez consulter Personnalisation de vos tables de clés KMS.



- Pour afficher uniquement les clés à région unique ou les clés multi-région dans votre tableau de clés, filtrez vos clés à l'aide de la propriété Regionality (Régionalité) de chaque clé. Pour obtenir de l'aide, veuillez consulter Tri et filtrage de vos clés KMS.



- Vous pouvez également trier et filtrer votre table Clés gérées par le client pour le préfixe d'ID de clé distinctif mrk-.

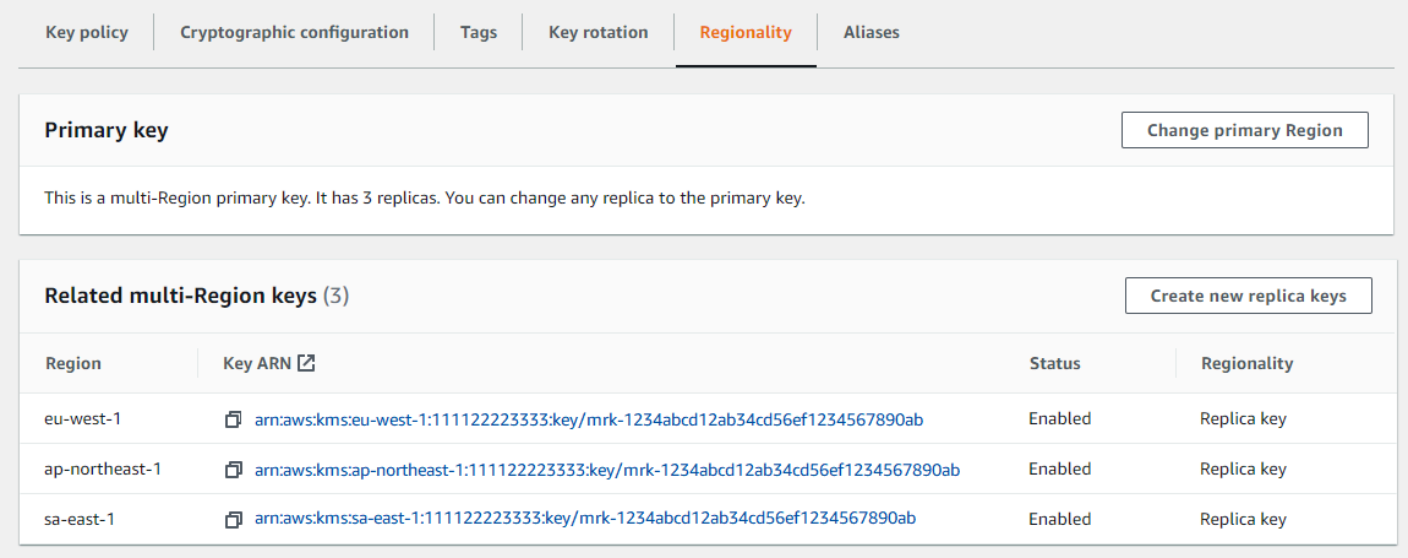


- Pour plus de détails sur une clé principale ou une clé de réplica multi-région, consultez la page détaillée de la clé, puis choisissez l'onglet Regionality (Régionalité).

  L'onglet Regionality (Régionalité) d'une clé principale inclut les boutons Change primary Region (Modifier la région principale) et Create new replica keys (Créer de nouvelles clés de réplica). (L'onglet Regionality (Régionalité) d'une clé de réplica n'a aucun bouton.) La section Related multi-Region keys (Clés multi-région associées) répertorie toutes les clés multi-région associées à la clé actuelle. Si la clé actuelle est une clé de réplica, cette liste inclut la clé principale.

  Si vous choisissez une clé multi-région associée dans le tableau Related multi-Region keys (Clés multi-région associées), la console AWS KMS passe à la région de la clé sélectionnée et ouvre la page détaillée de la clé. Par exemple, si vous choisissez la clé de réplica dans la région `sa-east-1` dans la section d'exemple Related multi-Region keys (Clés multi-région associées) ci-dessous, la console AWS KMS passe à la région `sa-east-1` pour afficher la page détaillée de cette clé de réplica. Vous pouvez le faire pour afficher l'alias ou la politique de clé de la clé de

réplica. Pour changer de région à nouveau, utilisez le Sélecteur de région dans l'angle supérieur droit de la page.



Affichage des clés multi-région dans l'API

Pour afficher les clés multirégionales dans l'AWS KMSAPI, utilisez l'DescribeKeyopération. Elle affiche la clé spécifiée et toutes ses clés multi-région associées.

Comme la console AWS KMS, les opérations d'API AWS KMS sont régionales. Par exemple, lorsque vous appelez les ListAliasesopérations ListKeysor, elles renvoient uniquement les ressources de la région actuelle ou spécifiée. Mais lorsque vous appelez l'opération `DescribeKey` sur une clé multi-région, la réponse inclut toutes les clés multi-région associées dans d'autres Régions AWS.

Par exemple, la demande `DescribeKey` suivante obtient des détails sur une clé de réplica multi-région d'exemple dans la région Asie-Pacifique (Tokyo) (`ap-northeast-1`).

```
$ aws kms describe-key \
        --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
        --region ap-northeast-1
```

La plupart des `KeyMetadata` dans la réponse décrivent la clé de réplica dans la région Asie-Pacifique (Tokyo), il s'agit du sujet de la demande. Cependant, l'élément `MultiRegionConfiguration` décrit la clé principale dans la région USA Ouest (Oregon) (us-west-2) et ses clés de réplica dans d'autres Régions AWS, y compris le réplica de la région Asie-

Pacifique (Tokyo). `DescribeKey` renvoie la même valeur `MultiRegionConfiguration` pour toutes les clés multi-régions associées.


```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1586329200.918,
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-northeast-1"
                },
                {
                    "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "sa-east-1"
```

```
                }
            ]
        }
    }
}
```

Gestion des clés multi-régions

Pour la plupart des actions, vous gérez les clés multi-région de la même manière que vous utilisez et gérez les clés à région unique. Vous pouvez activer et désactiver les clés, définir et mettre à jour des alias, des politiques de clé, des autorisations et des balises. Cependant, la gestion des clés multi-région diffère en ces points.

- Vous pouvez mettre à jour la région principale. Cela transforme l'une des clés de réplica en une clé principale, et la clé principale actuelle en un réplica.
- Vous gérez la rotation automatique des clés seulement sur la clé principale.
- Vous pouvez obtenir la clé publique pour une clé multi-région asymétrique à partir de l'une des clés principales ou de réplica associées.

La propriété multi-région que vous définissez lorsque vous créez une clé KMS est inaltérable. Vous ne pouvez pas convertir une clé à région unique en clé multi-région ou convertir une clé multi-région en clé à région unique.

Mise à jour de la région principale

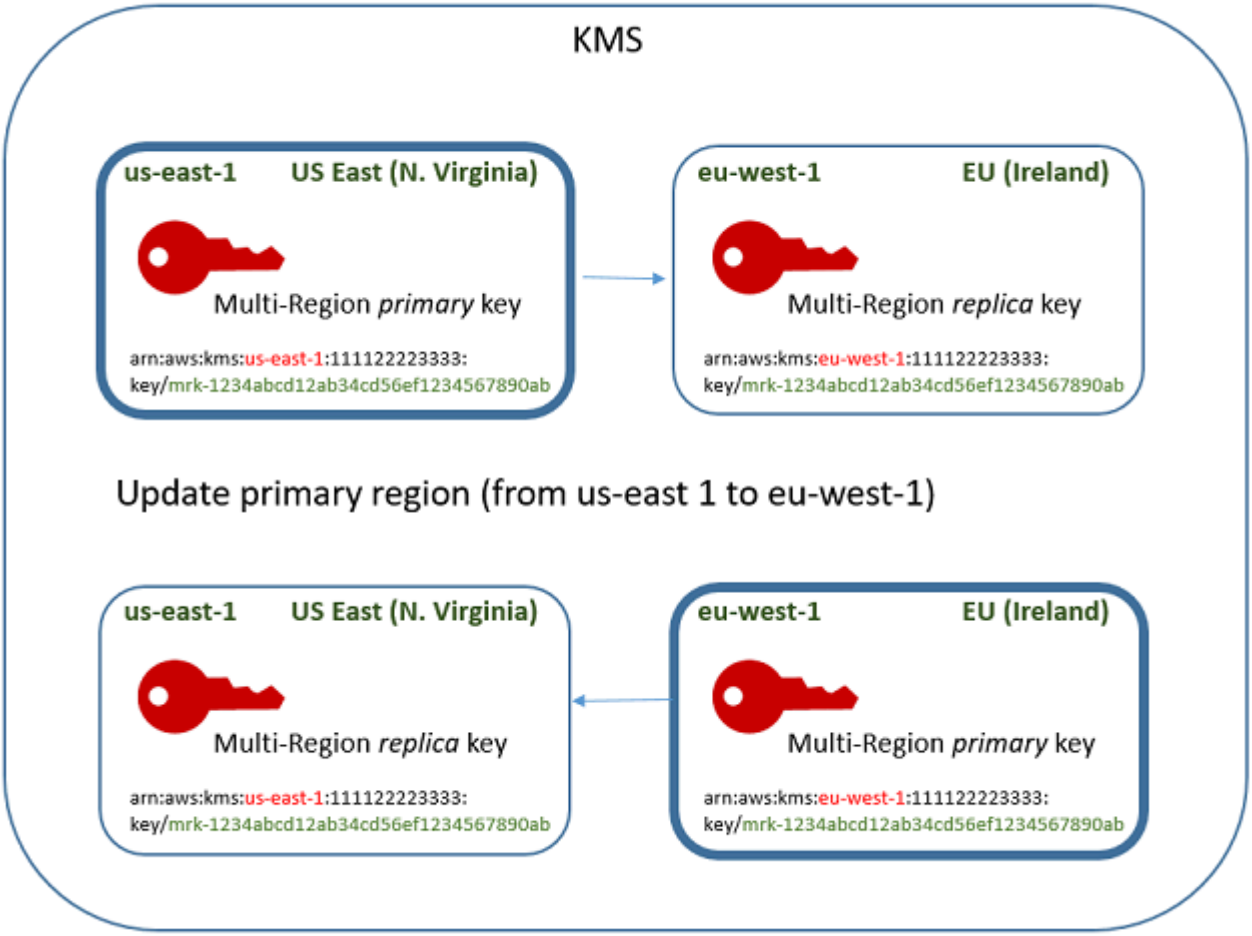
Chaque ensemble de clés multi-région associées doit posséder une clé principale. Mais vous pouvez changer la clé principale. Cette action, connue sous le nom de mise à jour de la région principale, convertit la clé principale actuelle en clé de réplica et convertit l'une des clés de réplica associées en clé principale. Vous pouvez le faire si vous avez besoin de supprimer la clé principale actuelle tout en conservant les clés de réplica, ou si vous devez localiser la clé principale dans la même région que vos administrateurs de clé.

Vous pouvez sélectionner n'importe quelle clé de réplica associée comme nouvelle clé principale. La clé principale et la clé de réplica doivent être toutes les deux dans l'état de clé `Enabled` lorsque l'opération démarre.

Même après la fin de cette opération, le processus de mise à jour de la région principale peut toujours être en cours pendant quelques secondes supplémentaires. Pendant ce temps, les anciennes et les nouvelles clés principales ont un état de clé transitoire Updating (Mise à jour en

cours). Lorsque l'état de la clé est `Updating`, vous pouvez utiliser les clés dans les opérations cryptographiques, mais vous ne pouvez pas répliquer la nouvelle clé principale ou effectuer certaines opérations de gestion, telles que l'activation ou la désactivation de ces clés. Des opérations telles que celles DescribeKeysusceptibles d'afficher à la fois les anciennes et les nouvelles clés primaires sous forme de répliques. L'état de la clé `Enabled` est restauré lorsque la mise à jour est terminée.

Supposons que vous avez une clé principale dans la région USA Est (Virginie du Nord) (us-east-1) et une clé de réplica dans la région UE (Irlande) (eu-west-1). Vous pouvez utiliser la fonction de mise à jour pour remplacer la clé principale dans la région USA Est (Virginie du Nord) (us-east-1) par une clé de réplica et transformer la clé de réplica dans la région UE (Irlande) (eu-west-1) par la clé principale.



Une fois le processus de mise à jour terminé, la clé multi-région dans la région UE (Irlande) (eu-west-1) est une clé principale multi-région et la clé dans la région USA Est (Virginie du Nord) (us-east-1) est sa clé de réplica. S'il existe d'autres clés de réplica associées, elles deviennent des réplicas de la nouvelle clé principale. La prochaine fois que AWS KMS synchronise les propriétés partagées des clés multi-région, il obtiendra les propriétés partagées à partir de la nouvelle clé principale et les copiera dans ses clés de réplica, y compris l'ancienne clé principale.

L'opération de mise à jour n'a aucun effet sur l'ARN de clé de n'importe quelle clé multi-région. Elle n'a pas non plus d'effet sur les propriétés partagées, telles que les éléments de clé, ni sur les propriétés indépendantes, telles que la politique de clé. Toutefois, vous devrez peut-être mettre à jour la politique de clé de la nouvelle clé principale. Par exemple, vous souhaiterez peut-être ajouter kms : ReplicateKey permission for trusted principals à la nouvelle clé primaire et la supprimer de la nouvelle clé de réplique.

L'état de clé **`Updating`**

Le processus de mise à jour d'une région principale prend un peu plus de temps que le bref délai de cohérence éventuelle qui affecte la plupart des opérations AWS KMS. Il se peut que le processus soit toujours en cours après que l'opération `UpdatePrimaryRegion` renvoie ses éléments, ou si vous avez terminé la procédure de mise à jour dans la console. Des opérations telles que l'DescribeKeyaffichage de l'ancienne et de la nouvelle clé primaire sous forme de répliques jusqu'à la fin du processus.

Au cours du processus de mise à jour de la région principale, l'ancienne clé principale et la nouvelle clé principale se trouvent à l'état de clé `Updating`. Lorsque le processus de mise à jour se termine avec succès, les deux clés retournent à l'état de clé `Enabled`. Lors de l'état `Updating`, certaines opérations de gestion, telles que l'activation et la désactivation des clés, ne sont pas disponibles. Toutefois, vous pouvez continuer à utiliser les deux clés dans les opérations cryptographiques sans interruption. Pour plus d'informations sur l'effet de l'état de clé `Updating`, veuillez consulter Principaux états des clés AWS KMS.

Mise à jour d'une région principale (console)

Vous pouvez mettre à jour la clé principale dans la console AWS KMS. Commencez sur la page des détails de la clé principale actuelle.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Sélectionnez l'ID de clé ou l'alias de la clé principale multi-région. La page des détails de la clé principale s'ouvre.

   Pour identifier une clé principale multi-région, utilisez l'icône de l'outil dans le coin supérieur droit pour ajouter la colonne Regionality (Régionalité) dans la table.

5. Cliquez sur l'onglet Regionality (Régionalité).
6. Dans la section Primary key (Clé principale), choisissez Change primary Region (Modifier la région principale).
7. Choisissez la région de la nouvelle clé principale. Vous ne pouvez choisir qu'une seule région dans le menu.

   Le menu Change primary Regions (Modifier les régions principales) n'inclut que les régions associées à une clé multi-région. Vous n'êtes peut-être pas autorisé à mettre à jour la région principale dans toutes les régions du menu.
8. Choisissez Change primary Region (Modifier la région principale).

Mise à jour d'une région principale (API AWS KMS)

Pour modifier la clé primaire dans un ensemble de clés multirégionales associées, utilisez l'UpdatePrimaryRegionopération.

Utilisez le paramètre `KeyId` pour identifier la clé principale actuelle. Utilisez le paramètre `PrimaryRegion` pour indiquer la Région AWS de la nouvelle clé principale. Si la clé principale n'a pas déjà de réplica dans la nouvelle région principale, l'opération échoue.

L'exemple suivant transforme la clé principale de la clé multi-région dans la région `us-west-2` en son réplica dans la région `eu-west-1`. Le paramètre `KeyId` identifie la clé principale actuelle dans la région `us-west-2`. Le paramètre `PrimaryRegion` spécifie la Région AWS de la nouvelle clé principale, `eu-west-1`.

```
$ aws kms update-primary-region \
      --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
      --primary-region eu-west-1
```

En cas de succès, cette opération ne renvoie aucune sortie ; seulement le code d'état HTTP. Pour voir l'effet, appelez l'DescribeKeyopération sur l'une des touches multirégions. Vous devrez peut-être attendre que l'état de la clé repasse à `Enabled`. Pendant que l'état de la clé est Updating (Mise à jour en cours), les valeurs de la clé peuvent toujours être en flux.

Par exemple, l'appel `DescribeKey` suivant obtient les détails sur la clé multi-région dans la région `eu-west-1`. La sortie indique que la clé multi-région dans la région `eu-west-1` est désormais la clé

principale. La clé multi-région associée (même ID de clé) dans la région `us-west-2` est désormais une clé de réplica.

```
$ aws kms describe-key \
      --key-id arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1609193147.831,
        "Enabled": true,
        "Description": "multi-region-key",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
           "MultiRegionKeyType": "PRIMARY",
           "PrimaryKey": {
              "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
              "Region": "eu-west-1"
           },
           "ReplicaKeys": [
              {
                 "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                 "Region": "us-west-2"
              }
           ]
        }
    }
}
```

Rotation de clés multi-région

Vous pouvez activer et désactiver la rotation automatique des éléments de clé pour les clés multi-région. La rotation automatique des clés est une propriété partagée des clés multi-région.

Vous activez et désactivez la rotation automatique des clés uniquement sur la clé principale.

- Quand AWS KMS synchronise les clés multi-région, il copie le paramètre de propriété de rotation des clés à partir de la clé principale vers toutes ses clés de réplica associées.
- Quand AWS KMS procède à la rotation des éléments de clé, il crée de nouveaux éléments de clé pour la clé principale, puis copie les nouveaux éléments de clé au-delà des limites de la région vers toutes les clés de réplica associées. Les éléments de clé ne quittent jamais AWS KMS non chiffrés. Cette étape est soigneusement contrôlée pour s'assurer que les éléments de clé sont entièrement synchronisés avant qu'une clé ne soit utilisée dans une opération cryptographique.
- AWS KMS ne chiffre aucune donnée avec les nouveaux éléments de clé tant que ces éléments ne sont pas disponibles dans la clé principale et chacune de ses clés de réplica.
- Lorsque vous répliquez une clé principale qui a fait l'objet d'une rotation, la nouvelle clé de réplica possède les éléments de clé actuels et toutes les versions précédentes des éléments de clé pour ses clés multi-région associées.

Ce modèle garantit que les clés multi-région associées sont entièrement interopérables. Toute clé multi-région peut déchiffrer tout texte chiffré par une clé multi-région associée, même si le texte chiffré a été chiffré avant la création de la clé.

La rotation automatique des clés n'est pas prise en charge sur les clés KMS asymétriques ou les clés KMS avec un élément de clé importé. Pour obtenir plus d'informations sur la rotation automatique des clés et des instructions pour l'activer et la désactiver, veuillez consulter Rotation des AWS KMS keys.

Téléchargement de clés publiques

Lorsque vous créez une clé KMS asymétrique multi-région, AWS KMS crée une paire de clés RSA ou ECC (courbe elliptique) pour la clé principale. Ensuite, il copie cette paire de clés sur chaque réplica de la clé principale. Par conséquent, vous pouvez télécharger la clé publique à partir de la clé principale ou de l'une de ses clés de réplica. Vous obtiendrez toujours les mêmes éléments de clé.

Pour plus d'informations sur le téléchargement et l'utilisation des clés publiques en dehors de AWS KMS, veuillez consulter Considérations particulières pour le téléchargement de clés publiques. Pour obtenir des instructions, consultez Téléchargement de clés publiques.

Importation des éléments de clé dans des clés multi-régions

Vous pouvez importer vos propres éléments de clé dans des clés KMS multi-région. Les clés multi-région que vous créez avec vos propres éléments de clé sont interopérables. Vous pouvez chiffrer des données dans une région et les déchiffrer dans n'importe quelle autre région avec une clé multi-région associée.

Cependant, vous devez gérer les éléments de clé.

- AWS KMS ne copie pas et ne synchronise pas les éléments de clé à partir d'une clé principale avec des éléments de clé importés vers ses clés de réplica. Vous devez importer les mêmes éléments de clé dans des clés associées principales et de réplica.
- Vous définissez le modèle d'expiration et les dates d'expiration de chaque clé indépendamment lorsque vous importez les éléments de clé. Vous pouvez configurer un modèle d'expiration et des dates d'expiration identiques ou différents pour les clés multi-région associées. Si les éléments de clé approchent de leur date d'expiration, vous devez les réimporter dans la clé multi-région concernée.

  Les états de clé des clés multi-région associées sont indépendants les uns des autres. Par exemple, si les éléments de clé de la clé principale expirent, ses clés de réplica ne sont pas affectées.

Les mêmes exigences de région pour les clés de réplica s'appliquent aux clés multi-région avec des éléments de clé importés. Si vous importez les mêmes éléments de clé dans des clés à région unique ou des clés multi-région non associées, ces clés KMS ne sont pas interopérables.

Vous pouvez créer des clés multi-région avec des éléments de clé importés symétriques, asymétriques ou HMAC. AWS KMS ne prend pas en charge le matériel clé importé dans les magasins de clés personnalisés. De plus, vous ne pouvez pas activer la rotation automatique des clés pour une clé KMS dont les éléments de clé sont importés.

Outre leurs fonctions multi-région, les clés multi-région avec des éléments de clé importés sont les mêmes que les autres clés KMS avec des éléments de clé importés. Pour obtenir des informations détaillées sur la création et la configuration de clés à région unique avec des éléments de clé importés, veuillez consulter À propos des clés importées.

Rubriques

- Pourquoi toutes les clés KMS avec des éléments de clé importés ne sont-elles pas interopérables ?

- Création d'une clé principale avec des éléments de clé importés
- Création d'une clé de réplica avec des éléments de clé importés

Pourquoi toutes les clés KMS avec des éléments de clé importés ne sont-elles pas interopérables ?

Les clés KMS à région unique avec des éléments de clé importés ne sont pas interopérables, même lorsqu'elles ont les mêmes éléments de clé. Quand AWS KMS utilise une clé KMS pour chiffrer les données, il lie de manière cryptographique certaines des métadonnées de clé au texte chiffré. Cela sécurise le texte chiffré de sorte que seule la clé KMS qui a chiffré des données peut les déchiffrer.

Les clés multi-région sont conçues pour être interopérables. En plus d'avoir les mêmes éléments de clé, ils ont un ID de clé et d'autres métadonnées identiques. Ainsi, les textes chiffrés qu'ils génèrent peuvent être déchiffrés par n'importe quelle clé multi-région associée. Par conséquent, les propriétés d'approbation des clés multi-région sont différentes de celles des clés à région unique. Mais pour certains clients, le bénéfice du déchiffrement dans plusieurs régions l'emporte sur la valeur de sécurité d'un texte chiffré dépendant d'une seule clé KMS dans une seule Région AWS.

Création d'une clé principale avec des éléments de clé importés

Pour créer une clé principale avec des éléments de clé importés, vous commencez par créer une clé KMS sans éléments de clé. Lorsque vous créez la clé primaire sans élément de clé, vous devez spécifier la spécification de clé qui reflète le type d'élément de clé que vous prévoyez d'importer. Ensuite, importez vos éléments de clé dans la clé principale.

La procédure de création d'une clé principale multi-région sans éléments de clé est presque identique à la procédure de création d'une clé à région unique sans éléments de clé. La seule différence est que vous spécifiez que la clé est une clé multi-région.

Les autorisations pour créer une clé primaire multirégionale avec du matériel clé importé sont les mêmes que celles requises pour créer une clé primaire multirégionale avec du matériel AWS KMS clé, y compris les CreateServiceLinkedRole autorisations kms : CreateKey et iam : dans une politique IAM. Vous pouvez utiliser les clés de KeyOrigin condition kms : MultiRegionKeyType et kms : pour autoriser ou refuser l'autorisation de créer des clés primaires multirégionales avec du matériel clé importé.

Lors de la création d'une clé primaire avec des éléments de clé importés dans la console AWS KMS, utilisez les paramètres de la section Advanced options (Options avancées). Vous ne pouvez pas modifier ces propriétés après la création de la clé KMS.

- Définissez Key material origin (Origine des éléments de clé) sur External (Import key material) (Externe (Importation d'éléments de clé)).
- Définir Multi-Region replication (Réplication multi-région) sur Allow this key to be replicated into other Regions (Autoriser cette clé à être répliquée dans d'autres régions).

Lorsque vous utilisez l'CreateKeyopération pour créer une clé primaire avec du matériel clé importé, utilisez les `MultiRegion` paramètres `Origin` et et spécifiez le `KeySpec` et le`KeyUsage`. L'exemple suivant crée une clé KMS `EXTERNAL` qui permet d'importer un élément de clé `ECC_NIST_P384`.

```
$ aws kms create-key --origin EXTERNAL --key-spec ECC_NIST_P384 --key-usage SIGN_VERIFY
 --multi-region
```

Le résultat est une clé principale multi-région sans éléments de clé et avec un état de clé de `PendingImport`.

Pour activer cette clé KMS, vous devez télécharger une clé publique et un jeton d'importation, utiliser la clé publique pour chiffrer vos éléments de clé, puis importer vos éléments de clé. Pour obtenir des instructions, veuillez consulter Importation de matériel clé pour les AWS KMS clés.

Création d'une clé de réplica avec des éléments de clé importés

Vous pouvez créer une clé de réplica multi-région dans la console AWS KMS ou à l'aide des opérations d'API AWS KMS. Pour répliquer une clé principale multi-région avec des éléments de clé importés, utilisez la même procédure que celle utilisée pour créer une clé de réplica avec des éléments de clé AWS KMS. Cependant, le résultat est différent. Au lieu de renvoyer une clé de réplica avec les mêmes éléments de clé que la clé principale, le processus de réplication renvoie une clé de réplica sans éléments de clé et avec un état de clé de `PendingImport`. Pour activer la clé de réplica, vous devez importer les mêmes éléments de clé dans la clé de réplica que vous avez importé dans sa clé principale.

Bien qu'il ne réplique pas les éléments de clé, AWS KMS crée la clé de réplica avec les mêmes ID de clé, spécifications de clé, utilisation de clé et origine des éléments de clé que la clé primaire. Il garantit également que les éléments de clé que vous importez dans la clé de réplica sont identiques aux éléments de clé que vous avez importés dans la clé principale.

Pour créer une clé de réplica avec des éléments de clé importés :

1. Créez une clé principale multi-région avec des éléments de clé importés.

2. Effectuez l'une des actions suivantes :

   Dans la console AWS KMS, choisissez une clé principale multi-région avec des éléments de clé importés. Puis, dans l'onglet Regionality (Régionalité), sélectionnez Create new replica keys (Créer de nouvelles clés de réplica). Pour obtenir des instructions, veuillez consulter Création de clés de réplica (console).

   Ou utilisez l'ReplicateKeyopération. Pour le paramètre `KeyId`, saisissez l'ID ou l'ARN de clé d'une clé principale multi-région avec des éléments de clé importés. Pour obtenir des instructions, veuillez consulter Création d'une clé de réplica (API AWS KMS).
3. Pour chaque nouvelle clé de réplica, suivez les étapes pour télécharger une clé publique et un jeton d'importation. Utilisez la clé publique pour chiffrer les éléments de clé de la clé principale, puis importez ces éléments dans la clé de réplica. Vous avez besoin d'une clé publique et d'un jeton d'importation différents pour chaque clé de réplica.

   Si les éléments de clé que vous tentez d'importer dans la clé de réplica sont différents des éléments de clé de la clé principale, l'opération échoue. AWS KMS ne requiert pas que le modèle d'expiration et les dates d'expiration soient coordonnés, mais vous devrez peut-être établir des règles métier pour vos clés multi-région. Pour obtenir des instructions, veuillez consulter Importation de matériel clé pour les AWS KMS clés.

Autorisations de répliquer des clés avec des éléments de clé importés

Pour créer une clé de réplica avec des éléments de clé importés, vous devez disposer des autorisations suivantes.

Dans la région de la clé principale :

- kms : ReplicateKey sur la clé primaire (dans la région de la clé primaire). Inclure cette autorisation dans la politique de clé principale ou dans une politique IAM.

Dans la région de la clé de réplica :

- kms : CreateKey dans une politique IAM.
- km : GetParametersForImport. Vous pouvez inclure cette autorisation dans la politique de clé de la clé de réplica ou dans une politique IAM.
- km : ImportKeyMaterial. Vous pouvez inclure cette autorisation dans la politique de clé de la clé de réplica ou dans une politique IAM.

- kms : TagResource est nécessaire pour attribuer des balises lors de la réplication. Incluez cette autorisation dans une politique IAM dans la région de réplica.
- kms : CreateAlias est nécessaire pour répliquer une clé dans la AWS KMS console. Pour plus d'informations, consultez Contrôle de l'accès aux alias.

Suppression de clés multi-régions

Lorsque vous n'utilisez plus de clé multi-région principale ou de réplica, vous pouvez planifier sa suppression.

Bien que la suppression de clés KMS soit toujours effectuée avec prudence, la suppression d'un réplica d'une clé multi-région est moins risquée, à condition que la clé principale existe toujours dans AWS KMS. Si vous supprimez une clé de réplica de sa région, mais que vous découvrez un texte chiffré qui a été chiffré sous la clé supprimée, vous pouvez déchiffrer ce texte chiffré avec n'importe quelle clé multi-région associée. Vous pouvez également recréer la clé de réplica en répliquant la clé principale dans la région de la clé de réplica.

Toutefois, la suppression d'une clé principale et de toutes ses clés de réplica est une opération très dangereuse, équivalente à la suppression d'une clé à région unique.

> **Warning**
>
> La suppression d'une clé KMS est destructrice et potentiellement dangereuse. Vous devez y avoir recours seulement lorsque vous êtes sûr de ne plus avoir besoin d'utiliser la clé KMS maintenant ni par la suite. Si vous n'en êtes pas sûr, vous devriez désactiver la clé KMS au lieu de la supprimer.

Pour supprimer une clé principale, vous devez d'abord supprimer toutes ses clés de réplica. Si vous devez supprimer une clé principale d'une région particulière sans supprimer ses clés de réplica, transformez la clé principale en clé de réplica en mettant à jour la région principale.

Avant de planifier la suppression d'une clé KMS, consultez les mises en garde présentées dans cette Suppression de AWS KMS keys rubrique et les rubriques qui expliquent comment déterminer l'utilisation passée d'une clé KMS et comment configurer une CloudWatch alarme vous avertissant de l'utilisation de la clé KMS pendant la période d'attente. Avant de supprimer la clé principale d'une clé multi-région asymétrique, veuillez consulter la rubrique Suppression de clés asymétriques.

Rubriques

- Autorisations de suppression de clés multi-région
- Comment supprimer une clé de réplica
- Comment supprimer une clé principale

Autorisations de suppression de clés multi-région

Pour planifier la suppression d'une clé multi-région, vous avez besoin uniquement de l'autorisation suivante.

- kms : ScheduleKeyDeletion — pour planifier la suppression de la clé multirégionale et définir sa période d'attente.

Nous vous recommandons également vivement de disposer des autorisations associées suivantes.

- kms : CancelKeyDeletion — pour annuler la suppression planifiée de la clé multirégionale.
- kms : DescribeKey — pour afficher l'état clé de la clé multirégionale et la liste des clés multirégionales associées.
- kms : DisableKey — pour vous donner la possibilité de désactiver une clé multirégionale au lieu de la supprimer.
- kms : EnableKey — pour restaurer la fonctionnalité d'une clé multirégionale après avoir annulé sa suppression.

Vous pouvez également inclure l'autorisation de répliquer et de modifier la clé principale.

- km : ReplicateKey
- km : UpdateReplicaRegion

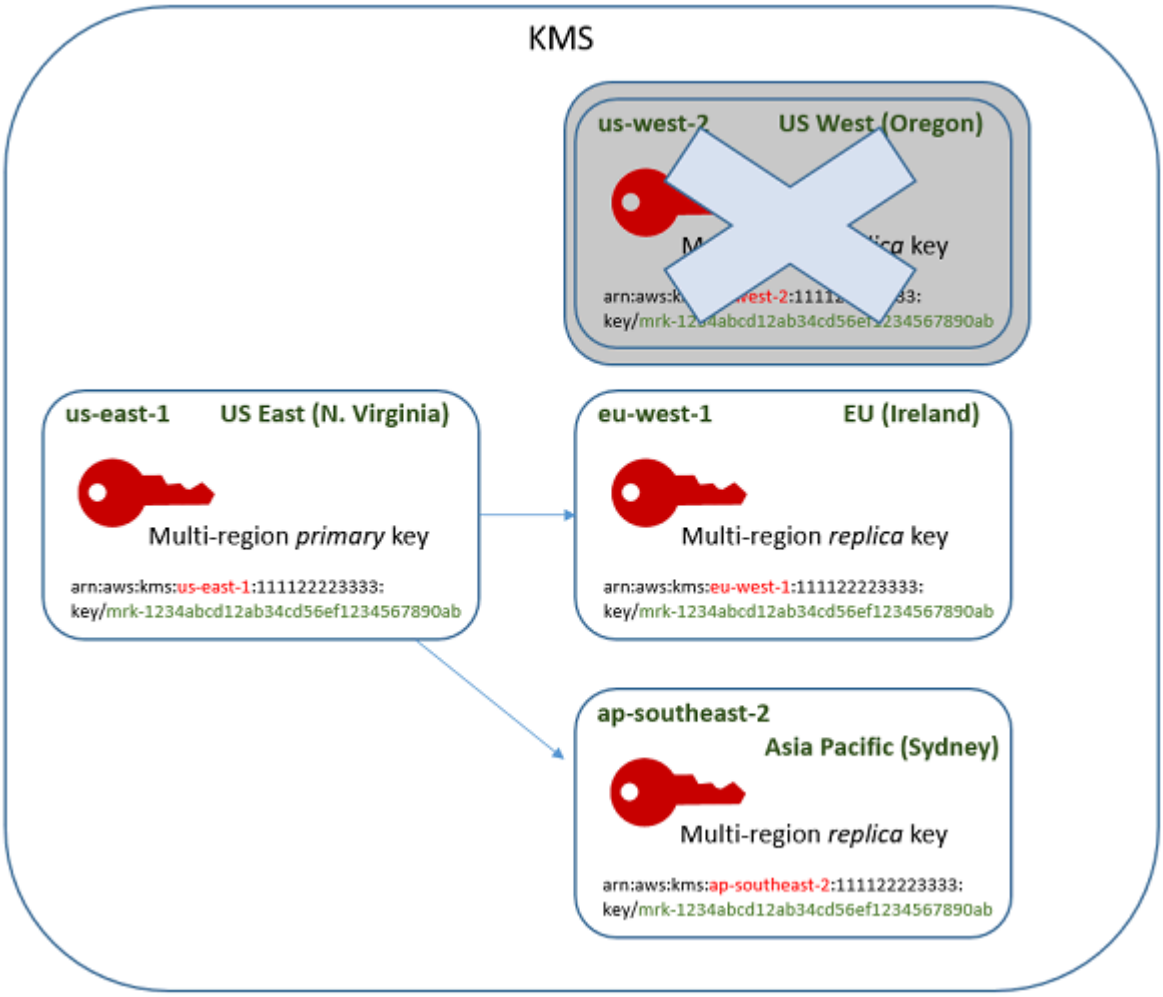
Vous pouvez inclure ces autorisations dans une politique IAM, mais la bonne pratique consiste à les placer dans une politique de clé où elles s'appliquent uniquement à la clé KMS que vous devez gérer.

Comment supprimer une clé de réplica

Vous pouvez utiliser la console AWS KMS ou l'API AWS KMS pour supprimer une clé de réplica. Vous pouvez supprimer une clé de réplica à tout moment. Cela ne dépend pas de l'état de clé d'une autre clé KMS.

Si vous supprimez par erreur une clé de réplica, vous pouvez la recréer en répliquant la même clé primaire dans la même région. La nouvelle clé de réplica que vous allez créer aura les mêmes propriétés partagées que la clé de réplica d'origine.

La procédure de suppression d'une clé de réplica multi-région est identique à celle de la suppression d'une clé à région unique.



1. Planifiez la suppression de la clé de réplica. Sélectionnez une période d'attente de 7 à 30 jours. La période d'attente par défaut est de 30 jours.
2. Pendant la période d'attente, l'état de clé de la clé de réplica passe à `Pending deletion` (`PendingDeletion`) et vous ne pouvez pas l'utiliser dans les opérations cryptographiques.
3. Vous pouvez annuler la suppression planifiée de la clé de réplica à tout moment de la période d'attente. L'état de la clé passe à `Disabled`, mais vous pouvez réactiver la clé KMS.
4. Lorsque la période d'attente expire, AWS KMS supprime la clé de réplica.

Vous pouvez afficher un enregistrement de vos actions dans votre journal AWS CloudTrail. AWS KMS enregistre les opérations qui planifient la suppression de la clé KMS et l'action qui supprime la clé KMS.

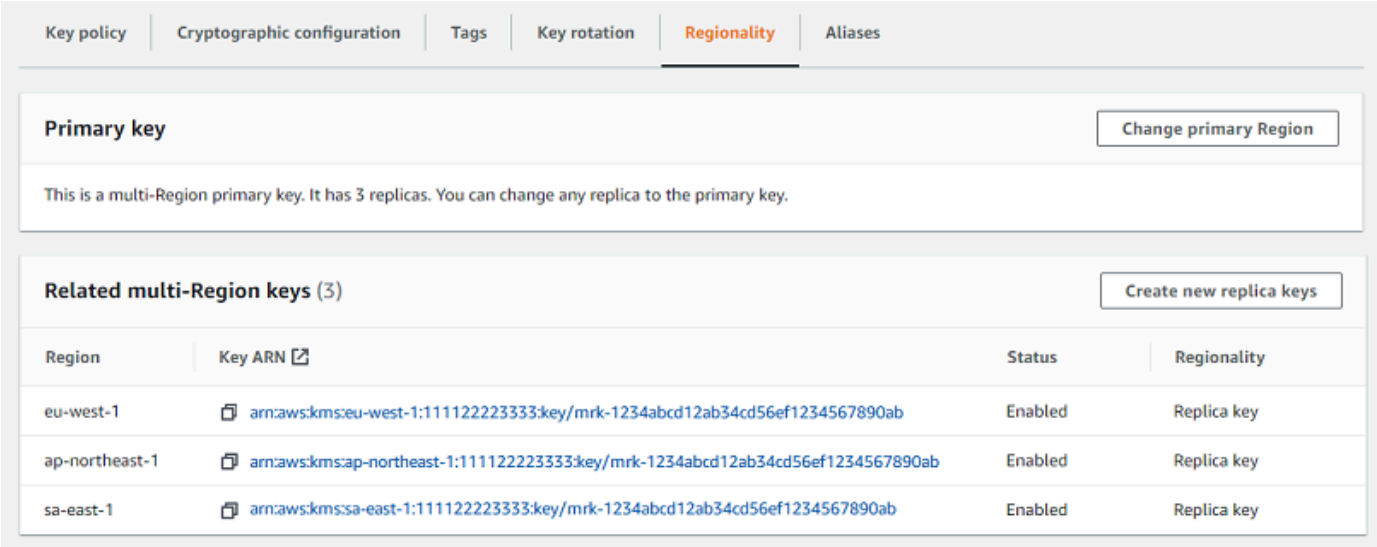
Suppression d'une clé de réplica (console)

Pour planifier la suppression d'une clé de réplica multi-région, utilisez la même procédure que vous utilisez pour planifier la suppression d'une clé à région unique.

Puisque les clés de réplica associées sont dans différentes Régions AWS, vous ne pouvez pas planifier la suppression de plusieurs clés de réplica à la fois. Pour supprimer toutes les clés de réplica associées, utilisez un modèle semblable au suivant.

Pour planifier la suppression de toutes les clés de réplica associées

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Dans le volet de navigation, sélectionnez Clés gérées par le client.
3. Utilisez le sélecteur de région dans l'angle supérieur droit pour choisir la région de la clé principale multi-région.
4. Choisissez l'alias ou l'ID de clé de la clé principale.
5. Cliquez sur l'onglet Regionality (Régionalité).



6. Dans la section Clés multi-région associées, choisissez l'ARN de clé d'une clé de réplica.

   Cette action ouvre la page de détails de la clé de réplica dans un nouvel onglet de navigateur. La console est définie sur la région de la clé de réplica.

7. Dans le menu Key actions (Actions de clé), sélectionnez Schedule key deletion (Planifier une suppression de clé).

   Cette action démarre le processus de planification de suppression de la clé. Terminez le processus de planification de suppression de la clé. Pour plus de détails, veuillez consulter Planification et annulation d'une suppression de clé (console).
8. Revenez à l'onglet du navigateur qui affiche l'onglet Regionality (Régionalité) de la clé principale. (Vous devrez peut-être actualiser la page pour voir l'état à jour des clés de réplica.) Choisissez l'ARN de clé d'une autre clé de réplica et répétez le processus de planification de suppression de la clé de réplica.

Suppression d'une clé de réplica (API AWS KMS)

Pour planifier la suppression d'une clé de réplique multirégionale, utilisez l'ScheduleKeyDeletionopération. Pour spécifier la clé KMS, utilisez son ID de clé ou son ARN de clé. Lorsque vous travaillez avec des clés multi-région, vous pouvez réduire l'incidence des erreurs en utilisant l'ARN de clé avec sa valeur de région explicite.

Par exemple, cette commande supprime une clé de réplica de la région us-west-2 (USA Ouest (Oregon)). Étant donné que la commande ne spécifie pas de période d'attente, la période d'attente est définie sur la valeur par défaut de 30 jours.

```
$ aws kms schedule-key-deletion \
    --region us-west-2 \
    --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Lorsque la commande aboutit, elle renvoie l'ARN de clé (`KeyId`), la période d'attente (`PendingWindowInDays`), la date de suppression (`DeletionDate`) et l'état actuel de la clé (`KeyState`), qui devrait être `PendingDeletion`.

Lorsque vous supprimez une clé de réplica multi-région, assurez-vous de vérifier que les valeurs d'ID de clé et de région dans l'ARN de clé sont celles que vous attendez.

```
{
    "KeyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "DeletionDate": 1599523200.0,
    "KeyState": "PendingDeletion",
```

```
    "PendingWindowInDays": 30
}
```

Pour supprimer tous les réplicas d'une clé principale multi-région par programmation, créez une liste des régions qui contiennent des clés de réplica. Ensuite, pour chaque région de la liste, appelez l'opération `ScheduleKeyDeletion`, comme indiqué ci-dessus.

Contrairement à une clé à région unique qui est définitivement supprimée, vous pouvez restaurer une clé de réplica en répliquant la clé principale dans la région où se trouvait la clé de réplica supprimée.

Pour vérifier l'état de la clé de réplique et afficher la clé primaire et les clés de réplique d'une clé multirégionale, utilisez l'DescribeKeyopération.

Comment supprimer une clé principale

Vous pouvez planifier la suppression d'une clé principale multi-région à tout moment. Toutefois, AWS KMS ne supprimera pas une clé principale multi-région qui possède des clés de réplica, même si leur suppression est planifiée.

Pour supprimer une clé principale, vous devez planifier la suppression de toutes ses clés de réplica, puis attendre que celles-ci soient supprimées. Le délai d'attente requis pour la suppression d'une clé principale commence lorsque la dernière de ses clés de réplica est supprimée. Si vous devez supprimer une clé principale d'une région particulière sans supprimer ses clés de réplica, transformez la clé principale en clé de réplica en mettant à jour la région principale.

Si une clé principale n'a pas de clés de réplica, le processus est identique à la suppression d'une clé de réplica ou d'une clé KMS régionale.

Lorsqu'une clé principale est programmée pour la suppression, vous ne pouvez pas l'utiliser dans les opérations cryptographiques et vous ne pouvez pas la répliquer. Toutefois, à moins que leur suppression ne soit également planifiée, ses clés de réplica ne sont pas affectées.

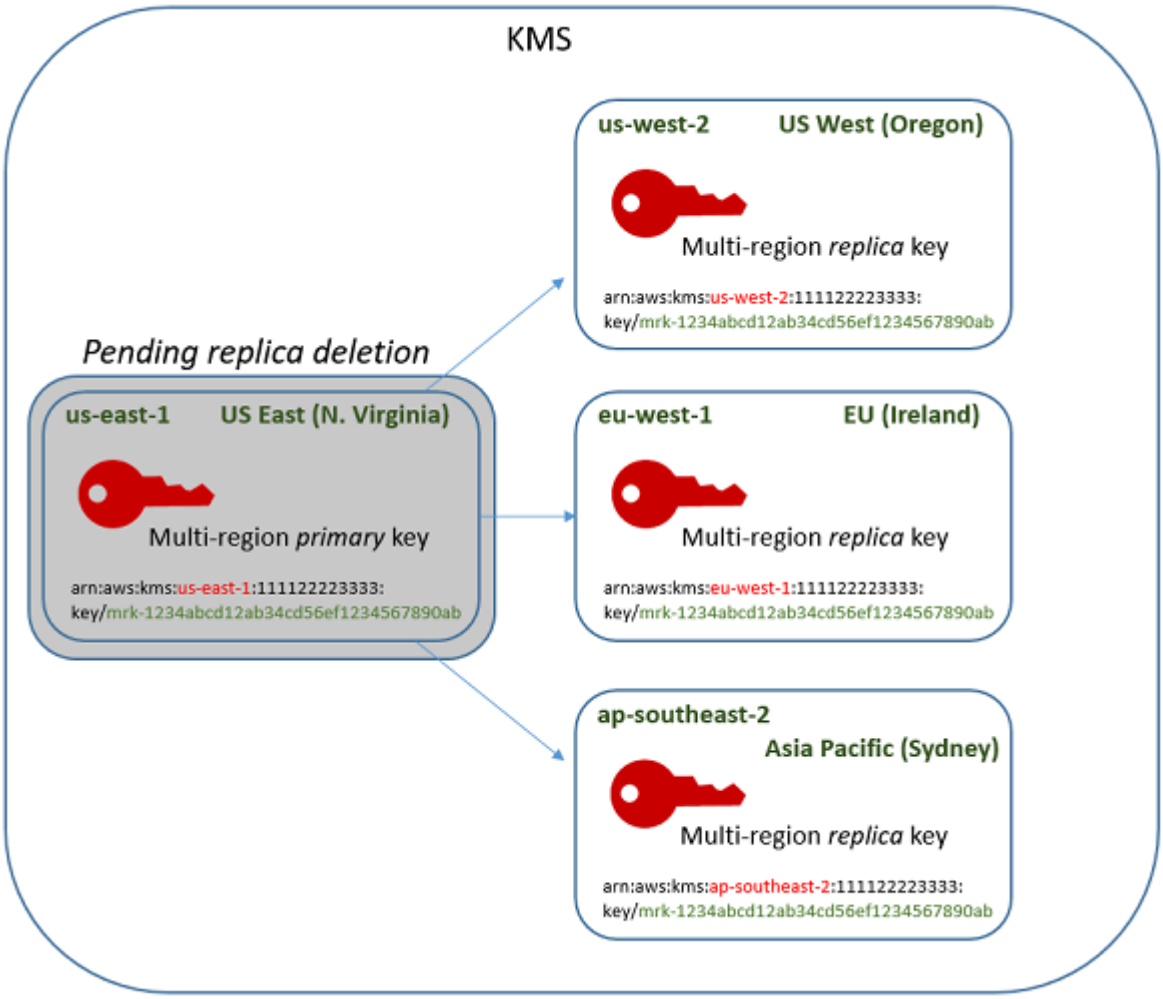
Vous pouvez utiliser la console AWS KMS ou l'API AWS KMS pour planifier la suppression des clés principales et de réplica. Vous pouvez planifier la suppression de la clé principale avant, après ou pendant que vous planifiez la suppression des clés de réplica. Le processus peut alors se présenter comme suit.

1. Planifiez la suppression de la clé principale. Sélectionnez une période d'attente de 7 à 30 jours. La période d'attente par défaut est de 30 jours. Toutefois, la période d'attente pour la clé principale ne commence pas tant que toutes les clés de réplica n'ont été supprimées.

S'il existe toujours des clés de réplica, l'état de clé de la clé principale passe à `Pending replica deletion` (`PendingReplicaDeletion`). Sinon, il passe à `Pending deletion` (`PendingDeletion`). Dans les deux cas, vous ne pouvez pas utiliser la clé principale dans les opérations cryptographiques et vous ne pouvez pas la répliquer.

La planification de la suppression d'une clé principale n'affecte pas les clés de réplica. Leur état de clé reste activé et vous pouvez les utiliser dans les opérations cryptographiques. Si les clés de réplica ne sont pas supprimées, l'état `Pending replica deletion` de la clé principale peut persister indéfiniment.

```
KMS key:                    Key state:
Primary (us-east-1)           Pending replica deletion (waiting period 30 days -- not
 started)
Replica (us-west-2)           Enabled
Replica (eu-west-1)           Enabled
Replica (ap-southeast-2)      Enabled
```



2. Planifiez la suppression de chaque clé de réplica. Sélectionnez une période d'attente de 7 à 30 jours. La période d'attente par défaut est de 30 jours. Vous pouvez supprimer plusieurs clés de réplica en même temps. Leurs délais d'attente se déroulent simultanément. Pendant le délai d'attente, l'état de clé des clés de réplica passe à `Pending deletion` (`PendingDeletion`) et vous ne pouvez pas les utiliser dans les opérations cryptographiques.

   Par exemple, si vous avez trois clés de réplica, vous pouvez planifier la suppression des trois en même temps. Elles peuvent avoir les mêmes délais d'attente ou des délais d'attente différents. Notez que le délai d'attente sur la clé principale n'a pas encore commencé. Son état de clé est `PendingReplicaDeletion`, car elle a des clés de réplica existantes.

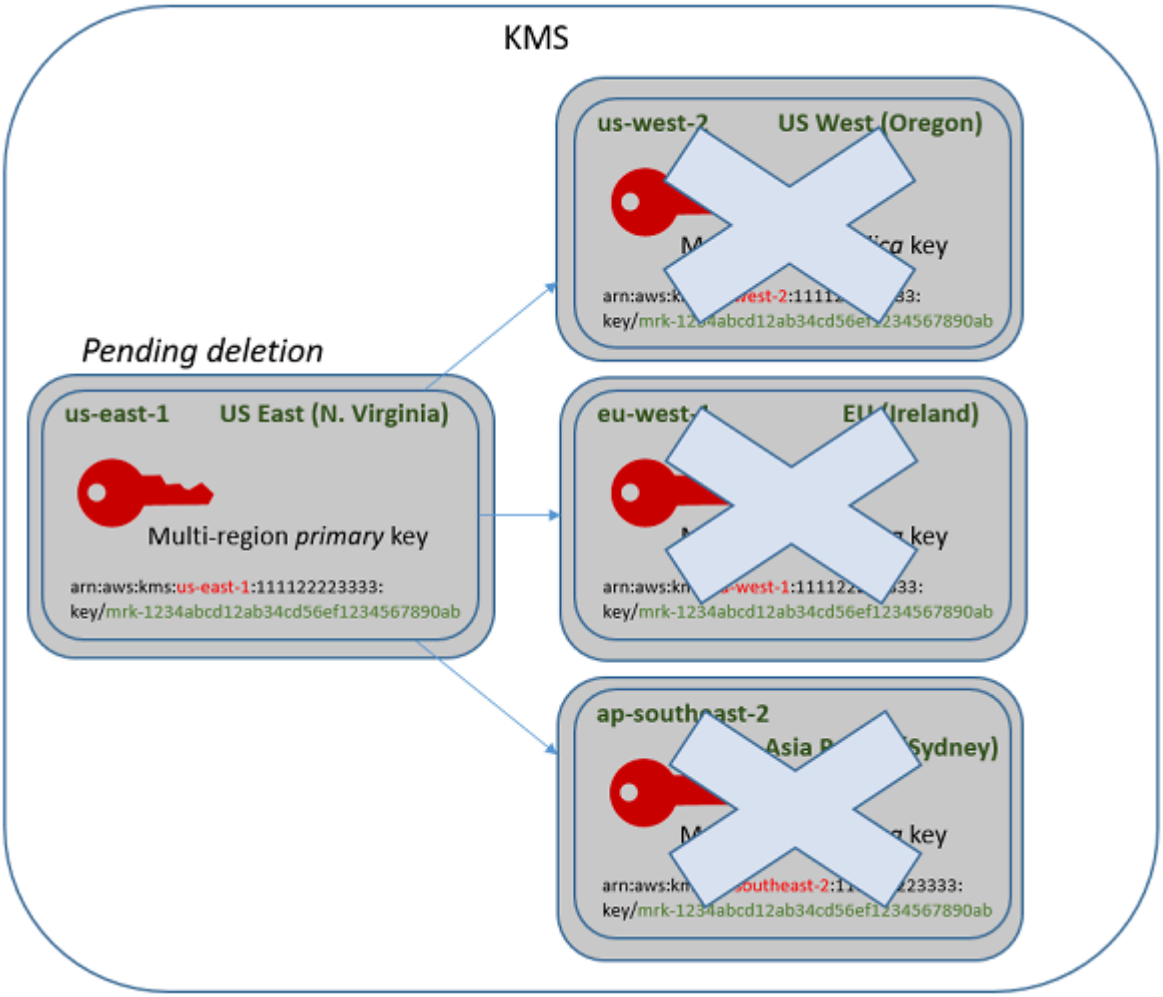
```
KMS key:                    Key state:
Primary key (us-east-1)       Pending replica deletion (waiting period 30 days -- not
 started)
Replica (us-west-2)           Pending deletion (7 days)
```

```
Replica (eu-west-1)           Pending deletion (7 days)
Replica (ap-southeast-2)      Pending deletion (30 days)
```

3. Vous pouvez annuler la suppression planifiée de la clé principale ou de toute clé de réplica jusqu'à ce qu'elle soit supprimée. L'état de la clé passe à `Disabled`, mais vous pouvez réactiver la clé KMS.

4. Lorsque la période d'attente de la dernière clé de réplica expire, AWS KMS supprime la dernière clé de réplica. L'état de clé de la clé principale passe de `Pending replica deletion` (`PendingReplicaDeletion`) à `Pending deletion` (`PendingDeletion`) et la période d'attente de 7 à 30 jours pour la clé principale commence.

```
KMS key:                    Key state:
Primary key (us-east-1)       Pending deletion (waiting period 30 days)
```



5. Lorsque sa période d'attente expire, AWS KMS supprime la clé principale.

Le délai minimal pour supprimer une clé primaire avec des réplicas est de 14 jours.

Si vous planifiez la suppression de la clé principale et de toutes les clés de réplica avec un délai d'attente de 7 jours, les clés de réplica sont supprimées au bout de 7 jours. La clé principale est supprimée le 14e jour.

- Jour 1 : planifiez la suppression des clés principales et de réplica avec une période d'attente minimale de 7 jours. Les périodes d'attente de suppression de 7 jours pour les clés de réplica démarrent. La période d'attente de suppression de la clé principale ne démarre pas encore.
- Jour 7 : les périodes d'attente de suppression des clés de réplica se terminent. AWS KMS supprime toutes les clés de réplica. Lorsque la dernière clé de réplica est supprimée, la période d'attente de suppression de 7 jours pour la clé principale démarre.
- Jour 14 : la période d'attente de suppression pour la clé principale se termine. AWS KMS supprime la clé principale.

Vous pouvez afficher un registre de vos actions dans votre journal AWS CloudTrail. AWS KMS enregistre les opérations qui planifient la suppression de chaque clé KMS et l'action qui supprime la clé KMS.

Suppression d'une clé principale (console)

Pour supprimer une clé principale multi-région, procédez comme suit.

Pour planifier une suppression de clé

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Cochez la case en regard de la clé principale que vous souhaitez supprimer. Vous pouvez également sélectionner une ou plusieurs clés KMS, y compris les réplicas de cette clé principale.
5. Choisissez Actions de clé, Planifier une suppression de clé.
6. Lisez et tenez compte de l'avertissement et des informations sur l'annulation de la suppression pendant la période d'attente. Si vous décidez d'annuler la suppression, choisissez Annuler.
7. Pour Période d'attente (en jours), tapez un nombre de jours compris entre 7 et 30. Si vous avez sélectionné plusieurs clés KMS, la période d'attente que vous choisissez s'applique à toutes les

clés KMS sélectionnées. La période d'attente pour les clés de réplica s'exécute simultanément, mais la période d'attente pour la clé principale ne commence pas avant que AWS KMS ne supprime la dernière des clés de réplica.

8. Cochez la case en regard pour confirmer que cette clé doit être supprimée en ***<nombre de jours>*** jours..
9. Choisissez Schedule deletion (Planifier la suppression).

Pour vérifier l'état de suppression de vos clés KMS, sur la page de détails de la clé principale, veuillez consulter la section General configuration (Configuration générale). L'état de la clé apparaît dans le champ Status (État). Lorsque l'état de clé de la clé principale passe à `Pending deletion`, la date de suppression planifiée s'affiche.

Vous pouvez également vérifier l'état de clé (Status (État)) de toutes les clés principales et de réplica dans l'onglet Regionality (Régionalité) de la page de détails d'une clé multi-région. Pour plus de détails, veuillez consulter Affichage des clés multi-régions.

Suppression d'une clé principale (API AWS KMS)

Pour supprimer une clé de réplique multirégionale, utilisez l'ScheduleKeyDeletionopération. Pour spécifier la clé KMS, utilisez son ID de clé ou son ARN de clé. Lorsque vous travaillez avec des clés multi-région, vous pouvez réduire l'incidence des erreurs en utilisant l'ARN de clé avec sa valeur de région explicite.

Par exemple, cette commande supprime une clé principale de la région us-east-1 (USA Est (Virginie du Nord). Étant donné que la commande ne spécifie pas de période d'attente, la période d'attente est définie sur la valeur par défaut de 30 jours.

```
$ aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Lorsque la commande aboutit, elle renvoie l'ARN de clé, l'état de la clé résultant et la période d'attente (`PendingWindowInDays`).

Si la clé principale n'a pas de réplicas, l'état de la clé principale est `PendingDeletion` et la sortie inclut le champ `DeletionDate`. Si des clés de réplica sont conservées, l'état de la clé principale est `PendingReplicaDeletion` et `DeletionDate` est omis, car il est incertain. Même si les clés de réplica sont également planifiées pour la suppression, vous pouvez annuler la suppression planifiée.

Lorsque vous supprimez une clé principale multi-région, assurez-vous de vérifier que les valeurs d'ID de clé et de région dans l'ARN de clé sont celles que vous attendez.

```
{
    "KeyId": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyState": "PendingReplicaDeletion",
    "PendingWindowInDays": 30
}
```

Pour vérifier l'état de suppression de vos clés KMS, utilisez l'DescribeKeyopération sur la clé primaire ou sur toute clé de réplique restante. La période d'attente pour la clé principale ne démarre pas tant que le dernier réplica n'est pas supprimé et que l'état de la clé n'est passé à `PendingDeletion`.

Pour calculer la date de suppression attendue de la clé principale, parcourez les ARN de clé de réplica dans la réponse, exécutez `DescribeKey` sur chacun d'eux, obtenez les dernières valeurs `DeletionDate`, puis ajoutez la valeur `PendingDeletionWindowInDays` pour la clé principale. Les périodes d'attente pour les clés de réplica s'exécutent simultanément.

Dans l'exemple suivant, la clé KMS est une clé principale multi-région avec des clés de réplica existantes. Puisque l'état de la clé est `PendingReplicaDeletion`, la réponse inclut la période d'attente (`PendingWindowInDays`), mais pas la `DeletionDate`. La date de suppression réelle de la clé principale dépend du moment où les clés de réplica sont supprimées.

```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1597902361.481,
        "Enabled": false,
        "Description": "",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingReplicaDeletion",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
```

```
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "us-west-2"
                },
{
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                }
            ]
        },
        "PendingDeletionWindowInDays": 30
    }
}
```

Lorsque tous les réplicas sont supprimés, la sortie `DescribeKey` affiche la clé principale restante avec un état de clé `PendingDeletion`. Alors que l'état de la clé est `PendingDeletion`, le champ `DeletionDate` apparaît à la place du champ `PendingWindowInDays`.

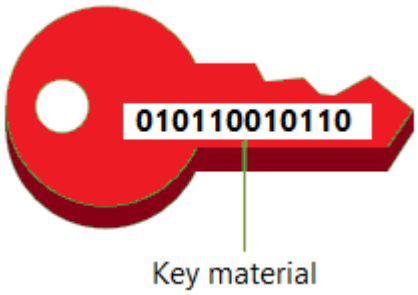
```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
```

```
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "",
        "CreationDate": 1597902361.481,
        "Enabled": false,
        "Description": "",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingDeletion",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "DeletionDate": 1597968000.0,
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": []
        }
    }
}
```

Importation de matériel clé pour les AWS KMS clés

Vous pouvez créer une AWS KMS keys (clé KMS) avec les éléments de clé que vous fournissez.

Une clé KMS est une représentation logique d'une clé de chiffrement. Les métadonnées d'une clé KMS incluent l'ID des éléments de clé utilisés pour chiffrer et déchiffrer des données. Lorsque vous créez une clé KMS, par défaut, AWS KMS génère les éléments de clé pour cette clé KMS. Mais vous pouvez créer une clé KMS sans éléments de clé, puis importer vos propres éléments de clé dans cette clé KMS, une fonction souvent appelée « Bring Your Own Key » (BYOK).



> **Note**
>
> AWS KMS ne prend pas en charge le AWS KMS déchiffrement de texte chiffré en dehors de AWS KMS, même si le texte chiffré a été chiffré sous une clé KMS avec du matériel clé importé. AWS KMS ne publie pas le format de texte chiffré requis par cette tâche, et le format peut changer sans préavis.

Le matériel clé importé est pris en charge sur tous les types de clés KMS, à l'exception des clés KMS dans les magasins de clés personnalisés. Toutefois, dans les régions de Chine, vous ne pouvez importer que des éléments de clé de chiffrement symétrique dans des clés KMS.

Lorsque vous utilisez du matériel clé importé, vous restez responsable du matériel clé tout en autorisant l'utilisation AWS KMS d'une copie de celui-ci. Vous pouvez choisir de le faire pour une ou plusieurs des raisons suivantes :

- Pour prouver que l'élément de clé a été généré en utilisant une source d'entropie correspondant à vos besoins.
- Pour utiliser le matériel clé de votre propre infrastructure avec AWS des services, et AWS KMS pour gérer le cycle de vie du matériel clé qu'il contient AWS.
- Pour utiliser des clés existantes et bien établies AWS KMS, telles que les clés pour la signature de code, la signature de certificats PKI et les applications associées à des certificats
- Pour définir une date d'expiration pour le contenu clé AWS et le supprimer manuellement, mais aussi pour le rendre à nouveau disponible à l'avenir. En revanche, une planification de suppression de clé nécessite une période d'attente de 7 à 30 jours, après laquelle vous ne pouvez pas récupérer la clé KMS supprimée.
- Posséder la copie originale du matériel clé et la conserver à l'extérieur AWS pour une durabilité accrue et une reprise après sinistre pendant tout le cycle de vie du matériau clé.

- Pour les clés asymétriques et les clés HMAC, l'importation crée des clés compatibles et interopérables qui fonctionnent à l'intérieur et à l'extérieur de. AWS

Vous pouvez auditer et surveiller l'utilisation et la gestion d'une clé KMS à l'aide de matériel clé importé. AWS KMS enregistre un événement dans votre AWS CloudTrail journal lorsque vous créez la clé KMS, que vous téléchargez la clé publique d'encapsulage et le jeton d'importation, et que vous importez le matériel clé. AWS KMS enregistre également un événement lorsque vous supprimez manuellement des éléments clés importés ou lorsque vous AWS KMS supprimez des éléments clés expirés.

Pour plus d'informations sur les différences importantes entre les clés KMS dont le contenu clé est importé et celles dont le contenu clé est généré par AWS KMS, voirÀ propos des clés importées.

Clés KMS prises en charge

AWS KMS prend en charge le matériel clé importé pour les types de clés KMS suivants. Vous ne pouvez pas importer des éléments de clé dans des clés KMS d'un magasin de clés personnalisé. Dans les régions de Chine, vous ne pouvez importer que des éléments de clé dans des clés de chiffrement symétrique.

- Clés KMS de chiffrement symétrique
- Clés KMS RSA asymétriques (pour le chiffrement ou la signature, mais pas les deux)
- Clés KMS de courbe elliptique asymétrique (ECC) (signature uniquement)
- Clés KMS HMAC
- Clés multi-région de tous les types pris en charge.

Régions

Le matériel clé importé est pris en charge dans tous Régions AWS les AWS KMS supports.

Dans les régions de Chine, vous ne pouvez importer que des éléments de clé dans des clés KMS de chiffrement symétrique. De plus, les exigences d'éléments de clé diffèrent de celles des autres régions. Pour plus de détails, consultez Importation des éléments de clé – Étape 3 : Chiffrement des éléments de clé.

Rubriques

- Planification pour importer des éléments de clé

- Gestion des éléments de clé importés
- Importation des éléments de clé Étape 1 : créer une AWS KMS key sans élément de clé
- Importation des éléments de clé – Étape 2 : Téléchargement de la clé publique d'encapsulage et du jeton d'importation
- Importation des éléments de clé – Étape 3 : Chiffrement des éléments de clé
- Importation des éléments de clé – Étape 4 : Importation des éléments de clé

Planification pour importer des éléments de clé

Le matériel clé importé vous permet de protéger vos AWS ressources à l'aide des clés cryptographiques que vous générez. L'élément de clé que vous importez est associé à une clé KMS particulière. Vous pouvez réimporter le même élément clé dans la même clé KMS, mais vous ne pouvez pas importer de contenu clé différent dans la clé KMS et vous ne pouvez pas convertir une clé KMS conçue pour le matériel clé importé en une clé KMS contenant du matériel AWS KMS clé.

En savoir plus :

- the section called "Sélectionnez une spécification de clé publique d'encapsulage"
- the section called "Sélectionner un algorithme d'encapsulage"

Rubriques

- À propos des clés importées
- Suppression des éléments de clé importés
- Autorisations d'importation des éléments de clé
- Exigences relatives aux éléments de clé importés

À propos des clés importées

Avant de décider d'importer du matériel clé dans AWS KMS, vous devez comprendre les caractéristiques suivantes du matériel clé importé.

Vous générez les éléments de clé.

> Vous êtes responsable de la génération des éléments de clé à l'aide d'une source aléatoire qui répond à vos exigences de sécurité.

Vous pouvez supprimer les éléments de clé.

Vous pouvez supprimer les éléments de clé importés d'une clé KMS, ce qui la rend immédiatement inutilisable. De plus, lorsque vous importez des éléments de clé dans une clé KMS, vous pouvez définir sa date d'expiration si vous souhaitez qu'elle en ait une. Lorsque le délai d'expiration arrive, AWS KMS supprime le matériel clé. Sans éléments de clé, la clé KMS ne peut pas être utilisée dans une opération de chiffrement. Pour restaurer la clé, vous devez y réimporter les mêmes éléments de clé.

Vous ne pouvez pas modifier les éléments de clé

Lorsque vous importez des éléments de clé dans une clé KMS, celle-ci est définitivement associée à ces éléments de clé. Vous pouvez réimporter les mêmes éléments de clé, mais vous ne pouvez pas en importer d'autres dans cette clé KMS. De plus, vous ne pouvez pas activer la rotation automatique des clés pour une clé KMS dont les éléments de clé sont importés. Toutefois, vous pouvez soumettre à une rotation manuelle une clé KMS avec les éléments de clé importés.

Vous ne pouvez pas modifier l’origine des éléments de clé

Les clés KMS conçues pour les éléments importés sont dotées d'une valeur origin (origine) non modifiable définie sur `EXTERNAL`. Vous ne pouvez pas convertir une clé KMS pour du matériel clé importé afin d'utiliser du matériel clé provenant d'une autre source, y compris AWS KMS. De même, vous ne pouvez pas convertir une clé KMS AWS KMS contenant du matériel clé en une clé conçue pour le matériel clé importé.

Vous ne pouvez pas exporter d’élément de clé

Vous ne pouvez pas exporter de matériel clé que vous avez importé. AWS KMS ne peut pas vous renvoyer le matériel clé importé sous quelque forme que ce soit. Vous devez conserver une copie du matériel clé importé à l'extérieur AWS, de préférence dans un gestionnaire de clés, tel qu'un module de sécurité matériel (HSM), afin de pouvoir réimporter le matériel clé si vous le supprimez ou s'il expire.

Vous pouvez créer des clés multi-région avec des éléments de clé importés

Les zones multirégionales avec un élément de clé importé ont les fonctionnalités des clés KMS avec un élément de clé importé et peuvent interagir entre Régions AWS. Pour créer une clé multi-région avec des éléments de clé importés, vous devez importer les mêmes éléments de clé dans la clé KMS principale et dans chaque clé de réplica. Pour plus de détails, consultez Importation des éléments de clé dans des clés multi-régions.

Les clés asymétriques et les clés HMAC sont portables et interopérables

Vous pouvez utiliser votre matériau de clé asymétrique et le matériau de clé HMAC à l'extérieur AWS pour interagir avec des AWS KMS clés contenant le même matériau de clé importé.

Contrairement au texte chiffré AWS KMS symétrique, qui est inextricablement lié à la clé KMS utilisée dans l'algorithme, AWS KMS utilise des formats HMAC standard et asymétriques pour le chiffrement, la signature et la génération de MAC. Par conséquent, les clés sont portables et prennent en charge les scénarios de clé sous séquestre traditionnels.

Lorsque votre clé KMS contient du matériel clé importé, vous pouvez utiliser le matériel clé importé AWS à l'extérieur pour effectuer les opérations suivantes.

- Clés HMAC – Vous pouvez vérifier une balise HMAC générée par la clé KMS HMAC avec un élément de clé importé. Vous pouvez également utiliser la clé HMAC KMS avec le matériel clé importé pour vérifier une étiquette HMAC qui a été générée par le matériel clé à l'extérieur de. AWS
- Clés de chiffrement asymétriques — Vous pouvez utiliser votre clé de chiffrement asymétrique privée AWS à l'extérieur pour déchiffrer un texte chiffré par la clé KMS avec la clé publique correspondante. Vous pouvez également utiliser votre clé KMS asymétrique pour déchiffrer un texte chiffré asymétrique généré en dehors de. AWS
- Clés de signature asymétriques — Vous pouvez utiliser votre clé KMS de signature asymétrique avec du matériel clé importé pour vérifier les signatures numériques générées par votre clé de signature privée en dehors de. AWSVous pouvez également utiliser votre clé de signature publique asymétrique AWS à l'extérieur pour vérifier les signatures générées par votre clé KMS asymétrique.

Si vous importez les mêmes éléments de clé dans différentes clés KMS du même Région AWS, ces clés sont également interopérables. Pour créer des clés KMS interopérables dans différentes régions Régions AWS, créez une clé multirégionale avec du matériel clé importé.

Les clés de chiffrement symétriques ne sont ni portables ni interopérables

Les textes chiffrés symétriques AWS KMS produits ne sont ni portables ni interopérables. AWS KMS ne publie pas le format de texte chiffré symétrique requis par la portabilité, et le format peut changer sans préavis.

- AWS KMS ne peut pas déchiffrer les textes chiffrés symétriques que vous chiffrez en dehors de ceux-ci AWS, même si vous utilisez des éléments clés que vous avez importés.

- AWS KMS ne prend pas en charge le déchiffrement d'un texte chiffré AWS KMS symétrique en dehors de AWS KMS, même si le texte chiffré a été chiffré sous une clé KMS avec du matériel clé importé.
- Les clés KMS avec le même élément de clé importé ne sont pas interopérables. Le texte chiffré symétrique qui AWS KMS génère un texte chiffré spécifique à chaque clé KMS. Ce format de texte chiffré garantit que seule la clé KMS qui a chiffré des données peut les déchiffrer.

En outre, vous ne pouvez utiliser aucun AWS outil, tel que le AWS Encryption SDKchiffrement côté client Amazon S3, pour déchiffrer AWS KMS des textes chiffrés symétriques.

Par conséquent, vous ne pouvez pas utiliser de clés contenant du matériel clé importé pour soutenir des accords d'entiercement de clés dans le cadre desquels un tiers autorisé ayant un accès conditionnel au contenu clé peut déchiffrer certains textes chiffrés en dehors de. AWS KMSPour prendre en charge le séquestre de clés, utilisez le kit AWS Encryption SDK pour chiffrer votre message sous une clé indépendante de AWS KMS.

Vous êtes responsable de la disponibilité et de la durabilité.

AWS KMS est conçu pour maintenir la haute disponibilité du matériel clé importé. Mais AWS KMS ne maintient pas la durabilité du matériau clé importé au même niveau que le matériau clé qui en AWS KMS génère. Pour plus de détails, consultez Suppression des éléments de clé importés.

Suppression des éléments de clé importés

Les éléments de clé que vous importez sont protégés pendant le transport et au repos. Avant d'importer le matériel clé, vous chiffrez (ou « enveloppez ») le contenu clé avec la clé publique d'une paire de clés RSA générée dans des modules de sécurité AWS KMS matériels (HSM) validés dans le cadre du programme de validation des modules cryptographiques FIPS 140-2. Vous pouvez chiffrer l'élément de clé directement avec la clé publique d'enveloppement, ou chiffrer l'élément de clé avec une clé symétrique AES, puis chiffrer la clé symétrique AES avec la clé publique RSA.

À réception, AWS KMS déchiffre le contenu de la clé avec la clé privée correspondante dans un AWS KMS HSM et le chiffre à nouveau sous une clé symétrique AES qui n'existe que dans la mémoire volatile du HSM. Votre élément de clé ne quitte jamais le HSM en texte brut. Il est déchiffré uniquement pendant son utilisation et uniquement dans les HSM. AWS KMS

L'utilisation de votre clé KMS avec l'élément de clé importé est déterminée uniquement par les politiques de contrôle d'accès que vous définissez sur la clé KMS. En outre, vous pouvez utiliser des

alias et des balises pour identifier et contrôler l'accès à la clé KMS. Vous pouvez activer et désactiver la clé, afficher et modifier ses propriétés, et la surveiller à l'aide de services tels que AWS CloudTrail.

Cependant, vous conservez la seule copie sûre de votre élément de clé. En échange de cette mesure de contrôle supplémentaire, vous êtes responsable de la durabilité et de la disponibilité globale du matériau clé importé. AWS KMS est conçu pour maintenir la haute disponibilité du matériel clé importé. Mais AWS KMS ne maintient pas la durabilité du matériau clé importé au même niveau que le matériau clé qui en AWS KMS génère.

Cette différence en durabilité est significative dans les cas suivants :

- Lorsque vous définissez une date d'expiration pour votre matériel clé importé, le AWS KMS matériel clé est supprimé après son expiration. AWS KMS ne supprime pas la clé KMS ni ses métadonnées. Vous pouvez créer une CloudWatch alarme Amazon qui vous avertit lorsque le matériel clé importé approche de sa date d'expiration.

  Vous ne pouvez pas supprimer le contenu clé AWS KMS généré pour une clé KMS et vous ne pouvez pas le définir AWS KMS comme expirant, bien que vous puissiez le faire pivoter.
- Lorsque vous supprimez manuellement le contenu clé importé, il AWS KMS supprime le contenu clé mais ne supprime pas la clé KMS ni ses métadonnées. En revanche, une planification de suppression de clé nécessite une période d'attente de 7 à 30 jours, après laquelle AWS KMS supprime définitivement la clé KMS, ses éléments de clé et ses métadonnées.
- Dans le cas peu probable de certaines défaillances régionales susceptibles de l'affecter AWS KMS (comme une perte totale de courant), vous AWS KMS ne pourrez pas restaurer automatiquement le matériel clé importé. Cependant, AWS KMS vous pouvez restaurer la clé KMS et ses métadonnées.

Vous devez conserver une copie du matériel clé importé à l'extérieur d' AWS un système que vous contrôlez. Nous vous recommandons de stocker une copie exportable des éléments de clé importés dans un système de gestion des clés, tel qu'un module de sécurité matérielle (HSM). Si l'élément de clé importé est supprimé ou expire, la clé KMS associée devient inutilisable tant que vous ne le réimportez pas. En cas de perte définitive des éléments de clé importés, tout texte chiffré au moyen de la clé KMS est irrécupérable.

Autorisations d'importation des éléments de clé

Pour créer et gérer des clés KMS avec des éléments de clé importés, l'utilisateur a besoin d'une autorisation pour les opérations de ce processus. Vous pouvez fournir

les autorisations `kms:GetParametersForImport`, `kms:ImportKeyMaterial` et `kms:DeleteImportedKeyMaterial` dans la politique de clé lorsque vous créez la clé KMS. Dans la AWS KMS console, ces autorisations sont ajoutées automatiquement pour les administrateurs de clés lorsque vous créez une clé avec une origine matérielle de clé externe.

Pour créer des clés KMS avec des éléments de clé importés, le principal a besoin des autorisations suivantes.

- kms : CreateKey (politique IAM)
  - Pour limiter cette autorisation aux clés KMS contenant du matériel clé importé, utilisez la condition de KeyOrigin politique kms : avec une valeur de`EXTERNAL`.

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- kms : GetParametersForImport (politique clé ou politique IAM)
  - Pour limiter cette autorisation aux demandes qui utilisent un algorithme d'encapsulage et une spécification de clé d'encapsulation particuliers, utilisez les conditions de WrappingKeySpec politique kms : WrappingAlgorithm et kms :.
- kms : ImportKeyMaterial (politique clé ou politique IAM)
  - Pour autoriser ou interdire le contenu clé qui expire et contrôler la date d'expiration, utilisez les conditions de ValidTo politique kms : ExpirationModel et kms :.

Pour réimporter du matériel clé importé, le principal a besoin des ImportKeyMaterial autorisations kms : GetParametersForImport et kms :.

Pour supprimer le matériel clé importé, le principal a besoin de l'DeleteImportedKeyMaterialautorisation kms :.

Par exemple, pour donner à l'exemple l'autorisation `KMSAdminRole` de gérer tous les aspects d'une clé KMS avec des éléments de clé importés, incluez une instruction de politique de clé telle que celle suivante dans la politique clé de la clé KMS.

```
{
  "Sid": "Manage KMS keys with imported key material",
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KMSAdminRole"
  },
  "Action": [
    "kms:GetParametersForImport",
    "kms:ImportKeyMaterial",
    "kms:DeleteImportedKeyMaterial"
  ]
}
```

Exigences relatives aux éléments de clé importés

L'élément de clé que vous importez doit être compatible avec les spécifications de clés de la clé KMS associée. Pour les paires de clés asymétriques, importez uniquement la clé privée de la paire. AWS KMS dérive la clé publique de la clé privée.

AWS KMS prend en charge les spécifications clés suivantes pour les clés KMS avec du matériel clé importé. Dans les régions de Chine, l'élément de clé importé n'est pris en charge que pour la spécification de clé SYMMETRIC_DEFAULT.

Spécification de clé KMS	Exigences relatives aux éléments de clés
Clés de chiffrement symétrique SYMMETRIC_DEFAULT	256 bits (32 octets) de données binaires Dans les régions de Chine, il doit s'agir de 128 bits (16 octets) de données binaires.
Clés HMAC HMAC_224 HMAC_256	L'élément de clé HMAC doit être conforme à la norme RFC 2104. La longueur de la clé doit correspondre à la longueur spécifiée par la spécification de clés.

Spécification de clé KMS	Exigences relatives aux éléments de clés
HMAC_384 HMAC_512	
Clé privée asymétrique RSA RSA_2048 RSA_3072 RSA_4096	La clé privée asymétrique RSA que vous importez doit faire partie d'une paire de clés conforme à la norme RFC 3447. Module : 2 048 bits, 3 072 bits ou 4 096 bits Nombre de nombres premiers : 2 (les clés RSA à plusieurs nombres premiers ne sont pas prises en charge) L'élément de clé asymétrique doit être codé en BER ou en DER au format PKCS (Public-Key Cryptography Standards) #8 conforme à la norme RFC 5208.
Clé privée asymétrique à courbe elliptique ECC_NIST_P256 (secp256r1) ECC_NIST_P384 (secp384r1) ECC_NIST_P521 (secp521r1) ECC_SECG_P256K1 (secp256k1)	La clé privée asymétrique ECC que vous importez doit faire partie d'une paire de clés conforme à la norme RFC 5915. Courbe : NIST P-256, NIST P-384, NIST P-521 ou Secp256k1 Paramètres : courbes nommées uniquement (les clés ECC avec des paramètres explicites sont rejetées) Coordonnées des points publics : peuvent être compressées, non compressées ou projectives L'élément de clé asymétrique doit être codé en BER ou en DER au format PKCS (Public-Key Cryptography Standards) #8 conforme à la norme RFC 5208.

Gestion des éléments de clé importés

Ces rubriques expliquent comment importer et réimporter des éléments clés dans une clé KMS et comment créer des éléments de clés importés qui expirent automatiquement.

Rubriques

- Vue d'ensemble de l'importation des éléments de clé
- Réimportation des éléments de clé
- Identification des clés KMS avec des éléments de clé importés
- Création d'une CloudWatch alarme en cas d'expiration du matériel clé importé
- Suppression des éléments de clé importés
- Suppression d'une clé KMS avec des éléments de clé importés

Vue d'ensemble de l'importation des éléments de clé

La présentation suivante explique comment importer vos éléments de clé dans AWS KMS. Pour plus d'informations sur chaque étape du processus, consultez la rubrique correspondante.

1. Créer une clé KMS sans élément de clé – L'origine doit être `EXTERNAL`. Une origine de clé `EXTERNAL` indique que la clé est conçue pour le matériel clé importé et AWS KMS empêche de générer du matériel clé pour la clé KMS. Dans une étape ultérieure, vous importerez vos propres éléments de clé dans cette clé KMS.

   Le matériel clé que vous importez doit être compatible avec les spécifications de la clé associée AWS KMS . Pour plus d'informations sur la compatibilité, veuillez consulter the section called “Exigences relatives aux éléments de clé importés”.
2. Téléchargement de la clé publique d’encapsulation et du jeton d'importation – Une fois l'étape 1 terminée, téléchargez une clé publique d'encapsulage et un jeton d'importation. Ces articles protègent votre matériel clé lorsqu'il est importé AWS KMS.

   Au cours de cette étape, vous choisissez le type (« spécification de clé ») de la clé d'encapsulation RSA et l'algorithme d'encapsulage que vous utiliserez pour chiffrer vos données en transit vers AWS KMS. Vous pouvez choisir une spécification de clé d'encapsulage et un algorithme de clé d'encapsulage différents chaque fois que vous importez ou réimportez le même élément de clé.

3. Chiffrement des éléments de clé – Utilisez la clé publique d'encapsulage que vous avez téléchargée à l'étape 2 pour chiffrer les éléments de clé que vous avez créés sur votre propre système.
4. Importation des éléments de clé – Téléchargez les éléments de clé chiffrés que vous avez créés à l'étape 3 et le jeton d'importation que vous avez téléchargé à l'étape 2.

   À ce stade, vous pouvez définir un délai d'expiration facultatif. Lorsque le contenu clé importé expire, il est AWS KMS supprimé et la clé KMS devient inutilisable. Pour continuer à utiliser la clé KMS, vous devez réimporter les mêmes éléments de clé.

   Lorsque l'opération d'importation est terminée, l'état de la clé KMS passe de `PendingImport` à `Enabled`. Vous pouvez désormais utiliser la clé KMS dans des opérations de chiffrement.

AWS KMS enregistre une entrée dans votre AWS CloudTrail journal lorsque vous créez la clé KMS, que vous téléchargez la clé publique d'encapsulage et le jeton d'importation, et que vous importez le matériel clé. AWS KMS enregistre également une entrée lorsque vous supprimez du matériel clé importé ou lorsque vous AWS KMS supprimez du matériel clé expiré.

Réimportation des éléments de clé

Si vous gérez une clé KMS avec des éléments de clé importés, vous pouvez avoir besoin de les réimporter. Vous pouvez réimporter des éléments de clé pour remplacer des éléments de clé expirés ou supprimés, ou pour modifier le modèle ou la date d'expiration de ceux-ci.

Lorsque vous importez des éléments de clé dans une clé KMS, celle-ci est définitivement associée à ces éléments de clé. Vous pouvez réimporter les mêmes éléments de clé, mais vous ne pouvez pas en importer d'autres dans cette clé KMS. Vous ne pouvez pas faire pivoter les éléments de clé et AWS KMS ne peut pas créer d'éléments de clé pour une clé KMS avec des éléments de clé importés.

Vous pouvez réimporter des éléments de clé à tout moment, selon une planification qui répond à vos exigences de sécurité. Vous n'avez pas besoin d'attendre que les éléments de clé arrivent à leur date d'expiration ou en soient proches.

Pour réimporter des éléments de clé, utilisez la même procédure que pour importer les éléments de clé la première fois, avec les exceptions suivantes.

- Utilisez une clé KMS existante au lieu de créer une nouvelle clé KMS. Vous pouvez ignorer l'étape 1 de la procédure d'importation.

- Lorsque vous réimportez des éléments de clé, vous pouvez modifier le modèle et la date d'expiration.

Chaque fois que vous importez des éléments de clé pour une clé KMS, vous devez télécharger et utiliser une nouvelle clé d'encapsulage et un nouveau jeton d'importation pour la clé KMS. La procédure d'encapsulage n'affecte pas le contenu de l'élément de clé. Vous pouvez ainsi utiliser différentes clés d'encapsulage et algorithmes d'encapsulation différents pour importer le même élément de clé.

Identification des clés KMS avec des éléments de clé importés

Lorsque vous créez une clé KMS sans éléments de clé, la valeur de la propriété `Origin` de la clé KMS est `EXTERNAL` et ne peut pas être modifiée. Contrairement à l'état de la clé, la valeur `Origin` ne dépend pas de la présence ou non d'éléments de clé.

Vous pouvez utiliser la valeur d'origine `EXTERNAL` pour identifier les clés KMS conçues pour les éléments clé importés. Vous pouvez trouver l'origine de la clé dans la AWS KMS console ou en utilisant l'DescribeKeyopération. Vous pouvez également afficher les propriétés des éléments de clé, par exemple leur date d'expiration éventuelle, à l'aide de la console ou des API.

Pour identifier les clés KMS avec des éléments de clé importés (console)

1. Ouvrez la AWS KMS console à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Utilisez l'une des techniques suivantes pour afficher la propriété `Origin` de vos clés KMS.
   - Pour ajouter une colonne d'origine à votre table de clé KMS. Dans le coin supérieur droit, choisissez l'icône Settings (Paramètres). Choisissez Origine, puis Confirmer. La colonne Origine permet d'identifier facilement les clés KMS avec une valeur de propriété d'origine Externe (Importation d'éléments de clé).
   - Pour rechercher la valeur de la propriété `Origin` d'une clé KMS particulière, choisissez l'ID de clé ou l'alias de la clé KMS. Choisissez ensuite l'onglet Cryptographic configuration (Configuration du chiffrement). Les onglets se trouvent sous la section General configuration (Configuration générale).
4. Pour consulter des informations détaillées sur les éléments de clé, sélectionnez l'onglet Key material (Éléments de clé). Cet onglet apparaît sur la page détaillée uniquement pour les clés KMS dont les éléments de clé sont importés.

Pour identifier les clés KMS avec du matériel clé importé (AWS KMS API)

Utilisez l'DescribeKeyopération. La réponse inclut la propriété `Origin` de la clé KMS, le modèle d'expiration et la date d'expiration, comme illustré dans l'exemple suivant.

```
$  aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyMetadata": {
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Origin": "EXTERNAL",
        "ExpirationModel": "KEY_MATERIAL_EXPIRES"
        "ValidTo": 2023-06-05T12:00:00+00:00,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": 2018-06-09T00:06:50.831000+00:00,
        "Enabled": false,
        "MultiRegion": false,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "PendingImport",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Création d'une CloudWatch alarme en cas d'expiration du matériel clé importé

Vous pouvez créer une CloudWatch alarme qui vous avertit lorsque le contenu clé importé d'une clé KMS approche de son expiration. Par exemple, l'alarme peut vous notifier lorsque le délai d'expiration est inférieur à 30 jours.

Lorsque vous importez des éléments de clé dans une clé KMS, vous pouvez éventuellement spécifier une date et heure à laquelle les éléments de clé doivent expirer. Lorsque le contenu clé expire, il est AWS KMS supprimé et la clé KMS devient inutilisable. Pour utiliser la clé KMS à nouveau, vous devez réimporter les éléments de clé. Toutefois, si vous réimportez les éléments de clé avant qu'ils n'expirent, vous pouvez éviter de perturber les processus qui utilisent cette clé KMS.

Cette alarme utilise la `SecondsUntilKeyMaterialExpires`métrique AWS KMS publiée sur CloudWatch pour les clés KMS dont le contenu clé importé expire. Chaque alarme utilise cette métrique pour surveiller les éléments de clé importés pour une clé KMS particulière. Vous ne pouvez pas créer une alarme unique pour toutes les clés KMS dont les éléments de clé expire ou une alarme pour les clés KMS que vous pourriez créer ultérieurement.

Prérequis

Les ressources suivantes sont requises pour une CloudWatch alarme qui surveille l'expiration du matériel clé importé.

- Une clé KMS dont les éléments de clé importés expirent. Pour obtenir de l'aide, veuillez consulter Identification des clés KMS avec des éléments de clé importés.
- Une rubrique Amazon SNS Pour plus de détails, consultez la rubrique Création d'un Amazon SNS dans le guide de CloudWatch l'utilisateur Amazon.

Créez l'alarme

Suivez les instructions de la section Création CloudWatch d'une alarme basée sur un seuil statique en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Sélectionner une métrique	Choisissez KMS, puis choisissez Per-Key Metrics. Choisissez la ligne contenant la clé KMS et la métrique `SecondsUntilKeyMaterialExpires` . Ensuite, choisissez Select metric (Sélectionner une métrique). La liste Métriques affiche les métriques `SecondsUntilKeyMaterialExpires` uniquement pour les clés KMS dont les éléments de clé importés expirent. Si vous ne disposez pas de clés KMS avec ces propriétés dans le compte et la région, cette liste est vide.
Statistique	Minimum
Période	1 minute

Champ	Valeur
Type de seuil	Statique
Chaque fois que …	Chaque fois que le *nom de la métrique* est supérieur à 1

Suppression des éléments de clé importés

Vous pouvez supprimer des éléments de clé importés d'une clé KMS à tout moment. De même, lorsque le matériel clé importé avec une date d'expiration expire, le AWS KMS matériel clé est supprimé. Dans les deux cas, lorsque l'élément de clé est supprimé, l'état de la clé de la clé KMS passe à en attente d'importation et celle-ci ne peut pas être utilisée dans le cadre des opérations de chiffrement avant de réimporter le même élément de clé. (Vous ne pouvez pas importer d'autres éléments de clé dans la clé KMS.)

Outre la désactivation de la clé KMS et le retrait des autorisations, la suppression des éléments de clé peut être utilisée comme stratégie pour arrêter rapidement, mais temporairement, l'utilisation de la clé KMS. En revanche, la planification de la suppression d'une clé KMS avec un élément de clé importé arrête également rapidement l'utilisation de la clé KMS. Toutefois, si la suppression n'est pas annulée pendant la période d'attente, la clé KMS, l'élément de clé et toutes les métadonnées de clés sont définitivement supprimés. Pour plus de détails, consultez the section called “Suppression d'une clé KMS avec des éléments de clé importés”.

Pour supprimer des éléments clés, vous pouvez utiliser la AWS KMS console ou l'opération DeleteImportedKeyMaterialAPI. AWS KMS enregistre une entrée dans votre AWS CloudTrail journal lorsque vous supprimez du matériel clé importé et lorsque vous AWS KMS supprimez du matériel clé expiré.

Rubriques

- Comment la suppression de documents clés affecte les AWS services
- Supprimer les éléments de clé (console)
- Supprimer le matériel clé (AWS KMS API)

Comment la suppression de documents clés affecte les AWS services

Lorsque vous supprimez des éléments de clé, la clé KMS sans éléments de clé devient immédiatement inutilisable (sous réserve d'une éventuelle cohérence). Toutefois, les ressources chiffrées à l'aide de clés de données protégées par la clé KMS ne sont pas affectées tant que la clé KMS n'est pas réutilisée, par exemple pour déchiffrer la clé de données. Ce problème concerne la Services AWSplupart d'entre eux qui utilisent des clés de données pour protéger vos ressources. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

Supprimer les éléments de clé (console)

Vous pouvez utiliser le AWS Management Console pour supprimer des éléments clés.

1. Connectez-vous à la console AWS Key Management Service (AWS KMS) AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Dans le volet de navigation, sélectionnez Clés gérées par le client.
4. Effectuez l’une des actions suivantes :

   - Cochez la case correspondant à une clé KMS avec les éléments de clé importés. Choisissez Key actions, Delete key material.
   - Choisissez l'alias ou l'ID de clé d'une clé KMS avec les éléments de clé importés. Cliquez sur l'onglet Key material (Éléments de clé), puis choisissez Delete key material (Suppression des éléments de clé).
5. Confirmez que vous souhaitez supprimer les éléments de clé, puis choisissez Delete key material. Le statut de la clé KMS, qui correspond à son état de clé, passe à Pending import (En attente d'importation).

Supprimer le matériel clé (AWS KMS API)

Pour utiliser l'AWS KMS API afin de supprimer du contenu clé, envoyez une DeleteImportedKeyMaterialdemande. L'exemple suivant montre comment procéder avec l'interface AWS CLI.

Remplacez `1234abcd-12ab-34cd-56ef-1234567890ab` par l'ID de clé de la clé KMS dont vous souhaitez supprimer les éléments de clé. Vous pouvez utiliser l'ID de clé ou le nom ARN de la clé KMS, mais vous ne pouvez pas utiliser un alias pour cette opération.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Suppression d'une clé KMS avec des éléments de clé importés

La suppression des éléments de clé d'une clé KMS avec des éléments de clé importés est temporaire et réversible. Pour restaurer la clé, réimportez son élément de clé.

En revanche, la suppression d'une clé KMS est irréversible. Si vous planifiez la suppression de la clé et que le délai d'attente requis expire, supprime AWS KMS définitivement et irréversiblement la clé KMS, son contenu clé et toutes les métadonnées associées à la clé KMS.

Cependant, les risques et les conséquences liés à la suppression d'une clé KMS contenant un élément de clé importé dépendent du type (« spécification de clé ») de la clé KMS.

- Clés de chiffrement symétriques – Si vous supprimez une clé KMS de chiffrement symétrique, tous les textes chiffrés restants chiffrés par cette clé sont irrécupérables. Vous ne pouvez pas créer une nouvelle clé KMS de chiffrement symétrique capable de déchiffrer les textes chiffrés d'une clé KMS de chiffrement symétrique supprimée, même si vous disposez du même élément de clé. Les métadonnées propres à chaque clé KMS sont liées par cryptographie à chaque texte chiffré symétrique. Cette fonctionnalité de sécurité garantit que seule la clé KMS qui a chiffré le texte chiffré symétrique peut le déchiffrer, mais elle vous empêche de recréer une clé KMS équivalente.
- Clés asymétriques et HMAC : si vous disposez du contenu de la clé d'origine, vous pouvez créer une nouvelle clé KMS avec les mêmes propriétés cryptographiques qu'une clé asymétrique ou HMAC KMS supprimée. AWS KMS génère des textes chiffrés et des signatures RSA standard, des signatures ECC et des balises HMAC, qui n'incluent aucune fonctionnalité de sécurité unique. Vous pouvez également utiliser une clé HMAC ou la clé privée d'une paire de clés asymétriques à l'extérieur de AWS.

  Une nouvelle clé KMS que vous créez avec le même élément de clé asymétrique ou HMAC aura un identifiant de clé différent. Vous devrez créer une nouvelle stratégie de clé, recréer tous les alias et mettre à jour les politiques et autorisations IAM existantes pour faire référence à la nouvelle clé.

Importation des éléments de clé Étape 1 : créer une AWS KMS key sans élément de clé

Par défaut, AWS KMS crée des éléments de clé pour vous lorsque vous créez une clé KMS. Pour importer vos propres éléments de clé à la place, commencez par créer une clé KMS sans élément de

clé. Procédez ensuite à l'importation. Pour créer une clé KMS sans élément clé, utilisez AWS KMS la console ou l'CreateKeyopération.

Pour créer une clé sans élément de clé, spécifiez l'origine de `EXTERNAL`. La propriété d'origine d'une clé KMS est immuable. Une fois créée, vous ne pouvez pas convertir une clé KMS conçue pour des éléments de clé importés en une clé pouvant recevoir des éléments provenant de AWS KMS ou toute autre source.

L'état d'une clé KMS avec une origine `EXTERNAL` et aucun élément de clé est `PendingImport`. Une clé KMS peut rester dans l'état `PendingImport` indéfiniment. Toutefois, vous ne pouvez pas utiliser une clé KMS dans l'état `PendingImport` dans le cadre des opérations cryptographiques. Lorsque vous importez l'élément de clé, l'état de la clé KMS passe à `Enabled`, et vous pouvez l'utiliser dans des opérations de chiffrement.

AWS KMSenregistre un événement dans votre AWS CloudTrail journal lorsque vous créez la clé KMS, que vous téléchargez la clé publique et le jeton d'importation, et que vous importez le contenu de la clé. AWS KMSenregistre également un CloudTrail événement lorsque vous supprimez du matériel clé importé ou lorsque vous AWS KMS supprimez du matériel clé expiré.

Pour plus d'informations sur la création de clés multi-région avec des éléments de clé importés, consultez Importation des éléments de clé dans des clés multi-régions.

Rubriques

- Création d'une clé KMS sans élément de clé (console)
- Création d'une clé KMS sans élément de clé (API AWS KMS)

Création d'une clé KMS sans élément de clé (console)

Vous devez créer uniquement une clé KMS pour les éléments de clé importés une seule fois. Vous pouvez importer et réimporter le même élément de clé sur la clé KMS existante aussi souvent que vous avez besoin, mais vous ne pouvez pas importer d'élément de clé différent dans une clé KMS. Pour plus de détails, consultez Étape 2 : Téléchargement de la clé publique d'encapsulage et du jeton d'importation.

Pour rechercher des clés KMS existantes contenant des éléments de clé importés dans votre tableau Customer managed keys (Clés gérées par le client), utilisez l'icône en forme d'engrenage dans le coin supérieur droit pour afficher la colonne Origin (Origine) de la liste des clés KMS. Les clés importées ont une valeur Origine égale à Externe (Importation des éléments de clé).

Pour créer une clé KMS avec un élément de clé importé, commencez par suivre les instructions de base pour créer une clé KMS du type de clé que vous préférez, à l'exception suivante.

Après avoir choisi l'utilisation de la clé, procédez comme suit :

1. (Facultatif) Développez Options avancées.
2. Dans le champ Key material origin (Origine des éléments de clé), sélectionnez External (Import key material) (Externe (Importation des éléments de clé)).
3. Cochez la case à côté de I understand the security, availability, and durability implications of using an imported key pour indiquer que vous comprenez les implications de l'utilisation d'éléments de clé importés. Pour de plus amples informations sur ces implications, veuillez consulter Suppression des éléments de clé importés.
4. Revenez aux instructions de base. Les étapes restantes de la procédure de base sont les mêmes pour toutes les clés KMS de ce type.

Lorsque vous choisissez Terminer, vous avez créé une clé KMS sans élément de clé et un statut (état de clé) En attente d'importation.

Cependant, au lieu de retourner au tableau des clés gérées par le client, la console affiche une page sur laquelle vous pouvez télécharger la clé publique et le jeton d'importation dont vous avez besoin pour importer l’élément de clé. Vous pouvez poursuivre l'étape de téléchargement dès maintenant ou choisir Annuler pour arrêter à ce stade. Vous pouvez revenir à cette étape de téléchargement à tout moment.

Suivant: Étape 2 : Téléchargement de la clé publique d’encapsulage et du jeton d'importation.

Création d'une clé KMS sans élément de clé (API AWS KMS)

Pour utiliser l'AWS KMSAPI afin de créer une clé KMS de chiffrement symétrique sans clé, envoyez une CreateKeydemande avec le `Origin` paramètre défini sur. `EXTERNAL` L'exemple suivant montre comment procéder avec l'AWS Command Line Interface (AWS CLI).

```
$ aws kms create-key --origin EXTERNAL
```

Lorsque la commande s'exécute correctement, vous obtenez une sortie similaire à ce qui suit. L'AWS KMS de la clé `Origin` est `EXTERNAL` et son `KeyState` est `PendingImport`.

> **Tip**
>
> Si la commande échoue, vous pouvez voir un `KMSInvalidStateException` ou un `NotFoundException`. Vous pouvez réessayer la demande.

```
{
    "KeyMetadata": {
        "Origin": "EXTERNAL",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "Enabled": false,
        "MultiRegion": false,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "PendingImport",
        "CreationDate": 1568289600.0,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Copiez la valeur `KeyId` dans la sortie de votre commande pour l'utiliser dans les étapes ultérieures, puis passez à l'Étape 2 : Téléchargement de la clé publique d’encapsulage et du jeton d'importation.

> **Note**
>
> Cette commande crée une clé KMS de chiffrement symétrique avec un `KeySpec` de `SYMMETRIC_DEFAULT` et `KeyUsage` de`ENCRYPT_DECRYPT`. Vous pouvez utiliser les paramètres facultatifs `--key-spec` et `--key-usage` créer une clé asymétrique ou KMS HMAC. Pour plus d'informations, consultez l'CreateKeyopération.

Importation des éléments de clé – Étape 2 : Téléchargement de la clé publique d’encapsulage et du jeton d'importation

Après avoir créé un élément AWS KMS key sans clé, téléchargez une clé publique encapsulante RSA et un jeton d'importation pour cette clé KMS à l'aide de la AWS KMS console ou de l'GetParametersForImportAPI. La clé publique d'encapsulage et le jeton d'importation constituent un ensemble indivisible qui doit être utilisé ensemble.

Vous utiliserez la clé publique d'encapsulage pour chiffrer votre élément de clé pour le transport. Avant le téléchargement, vous sélectionnez la longueur (spécification de clé) de la paire de clés d'encapsulage RSA et l'algorithme d'encapsulage que vous utiliserez pour chiffrer l’élément de clé importé en vue de son transport à l'étape 3.

Chaque ensemble de clés publiques d'encapsulage et de jetons d'importation est valide pendant 24 heures. Si vous ne les utilisez pas pour importer les éléments de clé dans les 24 heures après les avoir téléchargés, vous devez en télécharger de nouveaux. Vous pouvez télécharger de nouvelles clés publiques d'encapsulage et importer des ensembles de jetons à tout moment. Cela vous permet de modifier la longueur de votre clé d'encapsulage RSA (« spécification de clé ») ou de remplacer un ensemble perdu.

Vous pouvez également télécharger une clé publique d'encapsulage et un ensemble de jetons d'importation pour réimporter les mêmes éléments de clé dans une clé KMS. Vous pouvez le faire pour définir ou modifier le délai d'expiration des éléments de clé ou pour restaurer des éléments de clé expirés ou supprimés. Vous devez télécharger et rechiffrer votre élément de clé chaque fois que vous l'importez dans AWS KMS.

Utilisation de la clé publique d'encapsulage

Le téléchargement inclut une clé publique qui est unique à votre Compte AWS, également appelée une clé publique d’encapsulage.

Avant d'importer les éléments de clé, chiffrez les éléments de clé avec la clé publique d’encapsulage, puis charger les éléments de clé chiffrés vers AWS KMS. Lorsque AWS KMS reçoit votre élément de clé chiffré, il déchiffre l’élément de clé avec la clé privée correspondante, puis le chiffre à nouveau sous une clé symétrique AES, le tout dans un module de sécurité matériel (HSM) AWS KMS.

Utilisation du jeton d'importation

Le téléchargement comprend un jeton d'importation avec des métadonnées qui garantissent que vos éléments de clé sont importés correctement. Lorsque vous chargez vos éléments de clé chiffrés dans AWS KMS, vous devez charger le même jeton d'importation que vous avez téléchargé dans cette étape.

Sélectionnez une spécification de clé publique d'encapsulage

Pour protéger vos éléments de clé pendant l'importation, vous les chiffrez à l'aide de la clé publique d'encapsulage téléchargée sur AWS KMS et d'un algorithme d'encapsulage pris en charge. Vous sélectionnez une spécification de clé avant de télécharger votre clé publique d'encapsulage et votre jeton d'importation. Toutes les paires de clés d'encapsulage sont générées dans des modules de sécurité matériels (HSM) AWS KMS. La clé privée ne quitte jamais le HSM en texte brut.

La spécification de clé de la clé publique d'encapsulage détermine la longueur des clés de la paire de clés RSA qui protège votre élément de clé pendant son transport vers AWS KMS. En général, nous recommandons d'utiliser la clé publique d'encapsulage la plus longue qui soit pratique. Nous proposons plusieurs spécifications de clés publiques d'encapsulage pour prendre en charge une variété de HSM et de gestionnaires de clés.

AWS KMS prend en charge les spécifications de clés suivantes pour les clés d'encapsulage RSA utilisées pour importer des éléments de clé de tous types, sauf indication contraire.

- RSA_4096 (préféré)
- RSA_3072
- RSA_2048

> **Note**
>
> La combinaison suivante n'est PAS prise en charge : l'élément de clé ECC_NIST_P521, la spécification de clé d'encapsulage publique RSA_2048 et un algorithme d'encapsulage RSAES_OAEP_SHA_*.
> Vous ne pouvez pas directement encapsuler l'élément de clé ECC_NIST_P521 avec une clé d'encapsulage publique RSA_2048. Utilisez une clé d'encapsulage plus grande ou un algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_*.

Sélectionner un algorithme d'encapsulage

Pour protéger vos éléments de clé pendant l'importation, vous les chiffrez à l'aide de la clé publique d’encapsulage téléchargée et d'un algorithme d'enveloppement pris en charge.

AWS KMS prend en charge plusieurs algorithmes d'encapsulage RSA standard et un algorithme d'encapsulage hybride en deux étapes. En général, nous vous recommandons d'utiliser l'algorithme d'encapsulage le plus sécurisé qui soit compatible avec l’élément de clé importé et les spécifications de clé d’encapsulage. Généralement, vous choisissez un algorithme pris en charge par le module de sécurité matérielle (HSM) ou le système de gestion de clés qui protège vos éléments de clé.

Le tableau suivant présente les algorithmes d'encapsulage pris en charge pour chaque type d’élément de clé et de clé KMS. Les algorithmes sont répertoriés dans l'ordre de préférence.

Éléments de clé	Algorithme et spécification d'encapsulage pris en charge
Clés de chiffrement symétrique Clé AES 256 bits Clé SM4 128 bits (régions de Chine uniquement)	Algorithmes d'encapsulage : RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Algorithmes d'encapsulage obsolètes : RSAES_PKCS1_V1 **Note** Depuis le 10 octobre 2023, il AWS KMS ne prend pas en charge l'algorithme d'encapsulage RSAES_PKCS1_V1_5. Spécifications de clés d'encapsulage : RSA_2048 RSA_3072

Éléments de clé	Algorithme et spécification d'encapsulage pris en charge
	RSA_4096
Clé privée RSA asymétrique	Algorithmes d'encapsulage : RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 Spécifications de clés d'encapsulage : RSA_2048 RSA_3072 RSA_4096
Clé privée asymétrique à courbe elliptique (ECC) Vous ne pouvez pas utiliser les algorithmes d'encapsulage RSAES_OAEP_SHA_* avec la spécification de clé d'encapsulage RSA_2048 pour encapsuler l'élément de clé ECC_NIST_P521.	Algorithmes d'encapsulage : RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Spécifications de clés d'encapsulage : RSA_2048 RSA_3072 RSA_4096

Éléments de clé	Algorithme et spécification d'encapsulage pris en charge
Clé HMAC	Algorithmes d'encapsulage : RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Spécifications de clés d'encapsulage : RSA_2048 RSA_3072 RSA_4096

- `RSA_AES_KEY_WRAP_SHA_256` – Un algorithme d'encapsulage hybride en deux étapes qui combine le chiffrement de votre élément de clé avec une clé symétrique AES que vous générez, puis le chiffrement de la clé symétrique AES avec la clé d'encapsulage publique RSA téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_256.

  Un algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_* est requis pour encapsuler l'élément de clé privée RSA.
- `RSA_AES_KEY_WRAP_SHA_1` – Un algorithme d'encapsulage hybride en deux étapes qui combine le chiffrement de votre élément de clé avec une clé symétrique AES que vous générez, puis le chiffrement de la clé symétrique AES avec la clé publique d'encapsulage RSA téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_1.

  Un algorithme d'encapsulage `RSA_AES_KEY_WRAP_SHA_*` est requis pour encapsuler l'élément de clé privée RSA.
- `RSAES_OAEP_SHA_256` : algorithme de chiffrement RSA avec fonctionnalité OAEP (Optimal Asymmetric Encryption Padding) et fonction de hachage SHA-256.
- `RSAES_OAEP_SHA_1` : algorithme de chiffrement RSA avec fonctionnalité OAEP (Optimal Asymmetric Encryption Padding) et fonction de hachage SHA-1.
- `RSAES_PKCS1_V1_5` (Obsolète ; depuis le 10 octobre 2023, AWS KMS ne prend pas en charge l'algorithme d'encapsulage RSAES_PKCS1_V1_5) – Algorithme de chiffrement RSA avec format de remplissage défini dans PKCS #1 Version 1.5.

Rubriques

- Téléchargement de la clé publique d’encapsulage et du jeton d'importation (console)
- Téléchargement de la clé publique d’encapsulage et du jeton d'importation (API AWS KMS)

Téléchargement de la clé publique d’encapsulage et du jeton d'importation (console)

Vous pouvez utiliser la console AWS KMS pour télécharger la clé publique d’encapsulage et le jeton d'importation.

1. Si vous venez de terminer les étapes pour créer une clé KMS sans élément de clé et que vous êtes sur la page Download wrapping key and import token (Télécharger la clé d'encapsulage et le jeton d'importation), passez directement à Step 9.
2. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
3. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
4. Dans le volet de navigation, choisissez Clés gérées par le client.

   > **Tip**
   >
   > Vous pouvez uniquement importer un élément de clé dans une clé KMS d’une Origine Externe (Importation d'éléments de clé). Cela indique que la clé KMS a été créée sans élément de clé. Pour ajouter la colonne Origine à votre table, dans le coin supérieur droit de la page, choisissez l'icône des paramètres ( ). Activez Origine, puis choisissez Confirmer.

5. Choisissez l'alias ou l'ID de clé de la clé KMS qui est en attente d'importation.
6. Choisissez l'onglet Cryptographic configuration (Configuration de chiffrement) et affichez ses valeurs. Les onglets se trouvent sous la section General configuration (Configuration générale).

   Vous pouvez uniquement importer un élément de clé dans des clés KMS d’une Origine Externe (Importation d'éléments de clé). Pour plus d'informations sur la création de clés KMS avec des éléments de clé importés, veuillez consulter Importation de matériel clé pour les AWS KMS clés.
7. Choisissez l'onglet Key material (Éléments de clé) puis choisissez Import key material (Importation des éléments de clé).

L'onglet Key material (Éléments de clés) apparaît uniquement pour les clés KMS dont la valeur Origin (Origine) est définie sur (Externe (Importation d'éléments de clé).

8. Pour Sélectionner la spécification de la clé d'encapsulage, choisissez la configuration de votre clé KMS. Une fois cette clé créée, vous ne pouvez pas modifier ses spécifications.
9. Pour Sélectionner l'algorithme d'encapsulage, choisissez l'option que vous allez utiliser pour chiffrer vos éléments de clé. Pour de plus amples informations sur les options, veuillez consulter Sélectionner l'algorithme d'encapsulage.
10. Choisissez Télécharger la clé publique d'encapsulage et le jeton d'importation, puis enregistrez le fichier.

    Si vous avez une option Suivant, pour poursuivre le processus immédiatement, choisissez Suivant. Pour continuer ultérieurement, choisissez Annuler.
11. Décompressez le fichier `.zip` que vous avez enregistré à l'étape précédente (`Import_Parameters_<key_id>_<timestamp>`).

    Le dossier contient les fichiers suivants :

    - Une clé publique d'encapsulage RSA dans un fichier nommé `WrappingPublicKey.bin`.
    - Un jeton d'importation dans un fichier nommé `ImportToken.bin`.
    - Un fichier texte nommé README.txt. Ce fichier contient des informations sur la clé publique d'encapsulage, l'algorithme d'encapsulage à utiliser pour chiffrer vos éléments de clé, et la date et l'heure d'expiration de la clé publique d'encapsulage et du jeton d'importation.
12. Pour poursuivre le processus, consultez Chiffrer vos éléments de clé.

Téléchargement de la clé publique d'encapsulage et du jeton d'importation (API AWS KMS)

Pour télécharger la clé publique et le jeton d'importation, utilisez l'GetParametersForImportAPI. Spécifiez la clé KMS qui sera associée aux éléments de clé importé. Cette clé KMS doit avoir une Origin (Origine) définie à la valeur `EXTERNAL`.

Cet exemple indique l'algorithme d'encapsulage `RSA_AES_KEY_WRAP_SHA_256`, la spécification de clé publique d'encapsulage RSA_3072 et un exemple d'ID de clé. Remplacez ces exemples de valeurs par des valeurs valides pour votre téléchargement. Pour l'ID de clé, vous pouvez utiliser un ID de clé ou le nom ARN de clé, mais vous ne pouvez pas utiliser un nom d'alias ou un ARN d'alias pour cette opération.

```
$ aws kms get-parameters-for-import \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --wrapping-algorithm RSA_AES_KEY_WRAP_SHA_256 \
    --wrapping-key-spec RSA_3072
```

Lorsque la commande s'exécute correctement, vous obtenez une sortie similaire à ce qui suit :

```
{
    "ParametersValidTo": 1568290320.0,
    "PublicKey": "public key (base64 encoded)",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "ImportToken": "import token (base64 encoded)"
}
```

Pour préparer les données pour l'étape suivante, base64 décode la clé publique, importe le jeton et enregistre les valeurs décodées dans des fichiers.

Pour décoder la clé publique et le jeton d'importation en base64 :

1. Copiez la clé publique encodée en base64 (représentée par *clé publique (encodée en base64)* dans l'exemple de sortie), collez-la dans un nouveau fichier, puis enregistrez le fichier. Donnez au fichier un nom descriptif, par exemple `PublicKey.b64`.
2. Utilisez OpenSSL pour décoder du format base64 le contenu du fichier et enregistrer les données décodées dans un nouveau fichier. L'exemple suivant décode les données dans le fichier que vous avez enregistré à l'étape précédente (`PublicKey.b64`) et enregistre le résultat dans un nouveau fichier nommé `WrappingPublicKey.bin`.

   ```
   $ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
   ```

3. Copiez le jeton d'importation codé en base64 (représenté par *jeton d'importation (codé en base64)* dans l'exemple de sortie), collez-le dans un nouveau fichier, puis enregistrez le fichier. Donnez au fichier un nom descriptif, par exemple `importtoken.b64`.
4. Utilisez OpenSSL pour décoder du format base64 le contenu du fichier et enregistrer les données décodées dans un nouveau fichier. L'exemple suivant décode les données dans le fichier que vous avez enregistré à l'étape précédente (`ImportToken.b64`) et enregistre le résultat dans un nouveau fichier nommé `ImportToken.bin`.

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

Passez à Étape 3 : Chiffrement des éléments de clé.

Importation des éléments de clé – Étape 3 : Chiffrement des éléments de clé

Après avoir téléchargé la clé publique et le jeton d'importation, chiffrez l'élément de votre clé à l'aide de la clé publique que vous avez téléchargée et de l'algorithme d'encapsulage que vous avez spécifié. Si vous devez remplacer la clé publique ou le jeton d'importation, ou modifier l'algorithme d'encapsulage, vous devez télécharger une nouvelle clé publique et un nouveau jeton d'importation. Pour de plus amples informations sur les clés publiques et les algorithmes d'encapsulage qui prennent en charge AWS KMS, veuillez consulter Sélectionnez une spécification de clé publique d'encapsulage et Sélectionner un algorithme d'encapsulage.

La clé doit être au format binaire. Pour plus d'informations, consultez Exigences relatives aux éléments de clé importés.

> **Note**
>
> Pour les paires de clés asymétriques, chiffres et importez uniquement la clé privée. AWS KMSdérive la clé publique de la clé privée.
> La combinaison suivante n'est PAS prise en charge : l'élément de clé ECC_NIST_P521, la spécification de clé d'encapsulage publique RSA_2048 et un algorithme d'encapsulage RSAES_OAEP_SHA_*.
> Vous ne pouvez pas directement encapsuler l'élément de clé ECC_NIST_P521 avec une clé d'encapsulage publique RSA_2048. Utilisez une clé d'encapsulage plus grande ou un algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_*.

En règle générale, vous chiffrez vos éléments de clé lorsque vous les exportez à partir du module de sécurité matérielle (HSM) ou du système de gestion de clés. Pour plus d'informations sur l'exportation des clés au format binaire, consultez la documentation relative au module HSM ou au système de gestion de clés. Vous pouvez également vous reporter à la section suivante qui fournit la démonstration d'une preuve de concept à l'aide d'OpenSSL.

Lorsque vous chiffrez vos éléments de clé, utilisez le même algorithme d'encapsulage que celui que vous avez spécifié lorsque vous avez téléchargé la clé publique et le jeton d'importation. Pour trouver l'algorithme d'encapsulage que vous avez spécifié, consultez le CloudTrail journal des événements de la GetParametersForImportdemande associée.

Générer un élément de clé à tester

Les commandes OpenSSL suivantes génèrent des éléments de clés pour chaque type pris en charge à des fins de test. Ces exemples sont fournis uniquement à des fins de test et de proof-of-concept démonstration. Pour les systèmes de production, utilisez une méthode plus sécurisée pour générer votre élément de clé, notamment un module de sécurité matériel ou un système de gestion de clé.

Pour convertir les clés privées des paires de clés asymétriques au format codé DER, dirigez la commande de génération de l'élément de clé vers la commande `openssl pkcs8` suivante. Le paramètre `topk8` indique à OpenSSL de prendre une clé privée en entrée et de renvoyer une clé au format PKCS #8. (Le comportement par défaut est le contraire.)

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

Les commandes suivantes génèrent des éléments de clés de test pour chaque type de clé pris en charge.

- Clé de chiffrement symétrique (32 octets)

  Cette commande génère une clé symétrique de 256 bits (chaîne aléatoire de 32 octets) et l'enregistre dans le fichier `PlaintextKeyMaterial.bin`. Vous n'avez pas besoin d'encoder cet élément de clé.

  ```
  openssl rand -out PlaintextKeyMaterial.bin 32
  ```

  Dans les régions de Chine uniquement, vous devez générer une clé symétrique de 128 bits (chaîne aléatoire de 16 octets).

  ```
  openssl rand -out PlaintextKeyMaterial.bin 16
  ```

- Clés HMAC

  Cette commande génère une chaîne d'octets aléatoire de la taille spécifiée. Vous n'avez pas besoin d'encoder cet élément de clé.

La longueur de votre clé HMAC doit correspondre à la longueur définie par la spécification de clé de la clé KMS. Par exemple, si la clé KMS est HMAC_384, vous devez importer une clé de 384 bits (48 octets).

```
openssl rand -out HMAC_224_PlaintextKey.bin 28

openssl rand -out HMAC_256_PlaintextKey.bin 32

openssl rand -out HMAC_384_PlaintextKey.bin 48

openssl rand -out HMAC_512_PlaintextKey.bin 64
```

- Clés privées RSA

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:2048 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_2048_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:3072 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_3072_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_4096_PrivateKey.der
```

- Clés privées ECC

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-256 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P256_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-384 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P384_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-521 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P521_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:secp256k1 | openssl pkcs8 -
topk8 -outform der -nocrypt > ECC_SECG_P256K1_PrivateKey.der
```

Exemple de chiffrement des éléments de clé avec OpenSSL

Les exemples suivants montrent comment utiliser OpenSSL pour chiffrer votre élément de clé avec la clé publique que vous avez téléchargée.

> **Important**
>
> Ces exemples sont la démonstration d'une preuve de concept uniquement. Pour les systèmes de production, utilisez une méthode plus sécurisée (par exemple, un système de gestion de clés ou un module HSM commercial) pour générer et stocker vos éléments de clé.
> La combinaison suivante n'est PAS prise en charge : l'élément de clé ECC_NIST_P521, la spécification de clé d'encapsulage publique RSA_2048 et un algorithme d'encapsulage RSAES_OAEP_SHA_*.
> Vous ne pouvez pas directement encapsuler l'élément de clé ECC_NIST_P521 avec une clé d'encapsulage publique RSA_2048. Utilisez une clé d'encapsulage plus grande ou un algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_*.

RSAES_OAEP_SHA_1

AWS KMS prend en charge le RSAES_OAEP_SHA_1 pour les clés de chiffrement symétriques (SYMMETRIC_DEFAULT), les clés privées à courbe elliptique (ECC) et les clés HMAC.

RSAES_OAEP_SHA_1 n'est pas pris en charge pour les clés privées RSA. Vous ne pouvez pas non plus utiliser une clé publique d'encapsulage RSA_2048 avec un algorithme d'encapsulage RSAES_OAEP_SHA_* pour encapsuler une clé privée ECC_NIST_P521 (secp521r1).
Vous devez utiliser une clé d'encapsulage plus grande ou un algorithme d'encapsulage RSA_AES_KEY_WRAP.

L'exemple suivant chiffre votre élément de clé avec la clé publique que vous avez téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_1, puis l'enregistre dans le fichier `EncryptedKeyMaterial.bin`.

Dans cet exemple :

- *`WrappingPublicKey.bin`* est le fichier qui contient la clé publique d'encapsulage téléchargée.

- *PlaintextKeyMaterial.bin* est le fichier qui contient l'élément de clé que vous chiffres, tel que `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` ou `ECC_NIST_P521_PrivateKey.der`.

```
$ openssl pkeyutl \
    -encrypt \
    -in PlaintextKeyMaterial.bin \
    -out EncryptedKeyMaterial.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha1
```

RSAES_OAEP_SHA_256

AWS KMS prend en charge le RSAES_OAEP_SHA_256 pour les clés de chiffrement symétriques (SYMMETRIC_DEFAULT), les clés privées à courbe elliptique (ECC) et les clés HMAC.

RSAES_OAEP_SHA_256 n'est pas pris en charge pour les clés privées RSA. Vous ne pouvez pas non plus utiliser une clé publique d'encapsulage RSA_2048 avec un algorithme d'encapsulage RSAES_OAEP_SHA_* pour encapsuler une clé privée ECC_NIST_P521 (secp521r1). Vous devez utiliser une clé d'encapsulage plus grande ou un algorithme d'encapsulage RSA_AES_KEY_WRAP.

L'exemple suivant chiffre votre élément de clé avec la clé publique que vous avez téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_256, puis l'enregistre dans le fichier `EncryptedKeyMaterial.bin`.

Dans cet exemple :

- *WrappingPublicKey.bin* est le fichier qui contient la clé publique d'encapsulage téléchargée. Si vous avez téléchargé la clé publique à partir de la console, ce fichier est nommé `wrappingKey_`*KMS key_key_ID_timestamp* (par exemple, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`).
- *PlaintextKeyMaterial.bin* est le fichier qui contient l'élément de clé que vous chiffres, tel que `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` ou `ECC_NIST_P521_PrivateKey.der`.

```
$ openssl pkeyutl \
    -encrypt \
    -in PlaintextKeyMaterial.bin \
    -out EncryptedKeyMaterial.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha256 \
    -pkeyopt rsa_mgf1_md:sha256
```

RSA_AES_KEY_WRAP_SHA_1

L'algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_1 implique deux opérations de chiffrement.

1. Chiffrez votre élément de clé à l'aide d'une clé symétrique AES que vous générez et d'un algorithme de chiffrement symétrique AES.
2. Chiffrez la clé symétrique AES que vous avez utilisée avec la clé publique que vous avez téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_1.

AWS KMS prend en charge les algorithmes d'encapsulage RSA_AES_KEY_WRAP_SHA_* pour tous les types de documents clés importés pris en charge et toutes les spécifications de clés publiques prises en charge. Les algorithmes RSA_AES_KEY_WRAP_SHA_* sont les seuls algorithmes d'encapsulage pris en charge pour encapsuler l'élément de clé RSA.

L'algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_1 nécessite la version 3.x ou version ultérieure d'OpenSSL.

1. Générez une clé de chiffrement symétrique AES 256 bits

   Cette commande génère une clé de chiffrement symétrique AES composée de 256 bits aléatoires et l'enregistre dans le fichier `aes-key.bin`

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. Chiffrez votre élément de clé avec la clé de chiffrement symétrique AES

   Cette commande chiffre l'élément de votre clé avec la clé de chiffrement symétrique AES et enregistre le contenu de la clé chiffrée dans le fichier `key-material-wrapped.bin`.

   Dans cet exemple de commande :

   - *`PlaintextKeyMaterial.bin`* est le fichier qui contient l'élément de clé que vous importez, tel que `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der`, ou `ECC_NIST_P521_PrivateKey.der`.
   - *`aes-key.bin`* est le fichier qui contient la clé de chiffrement symétrique AES 256 bits que vous avez générée dans la commande précédente.

   ```
   # Encrypt your key material with the AES symmetric encryption key
   $ openssl enc -id-aes256-wrap-pad \
           -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
           -iv A65959A6 \
           -in PlaintextKeyMaterial.bin\
           -out key-material-wrapped.bin
   ```

3. Chiffrez votre clé de chiffrement symétrique AES avec la clé publique

   Cette commande chiffre votre clé de chiffrement symétrique AES avec la clé publique que vous avez téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_1, la code DER et l'enregistre dans le fichier `aes-key-wrapped.bin`.

   Dans cet exemple de commande :

   - *`WrappingPublicKey.bin`* est le fichier qui contient la clé publique d'encapsulage téléchargée. Si vous avez téléchargé la clé publique à partir de la console, ce fichier est nommé `wrappingKey_`*`KMS key_key_ID_timestamp`* (par exemple, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`)
   - *`aes-key.bin`* est le fichier qui contient la clé de chiffrement symétrique AES 256 bits que vous avez générée dans la première commande de cette séquence d'exemple.

   ```
   # Encrypt your AES symmetric encryption key with the downloaded public key
   $ openssl pkeyutl \
       -encrypt \
   ```

```
    -in aes-key.bin \
    -out aes-key-wrapped.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha1 \
    -pkeyopt rsa_mgf1_md:sha1
```

4. Générez le fichier à importer

   Concaténez le fichier contenant l'élément de clé chiffré et le fichier contenant la clé AES chiffrée. Enregistrez-les dans le fichier `EncryptedKeyMaterial.bin`, qui est le fichier que vous allez importer dans le Étape 4 : Importation des éléments de clé.

   Dans cet exemple de commande :

   - *key-material-wrapped.bin* est le fichier qui contient votre élément de clé chiffré.
   - *aes-key-wrapped.bin* est le fichier qui contient la clé de chiffrement AES chiffrée.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

RSA_AES_KEY_WRAP_SHA_256

L'algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_256 implique deux opérations de chiffrement.

1. Chiffrez votre élément de clé à l'aide d'une clé symétrique AES que vous générez et d'un algorithme de chiffrement symétrique AES.
2. Chiffrez la clé symétrique AES que vous avez utilisée avec la clé publique que vous avez téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_256.

AWS KMS prend en charge les algorithmes d'encapsulage RSA_AES_KEY_WRAP_SHA_* pour tous les types de documents clés importés pris en charge et toutes les spécifications de clés publiques prises en charge. Les algorithmes RSA_AES_KEY_WRAP_SHA_* sont les seuls algorithmes d'encapsulage pris en charge pour encapsuler l'élément de clé RSA.

L'algorithme d'encapsulage RSA_AES_KEY_WRAP_SHA_256 nécessite la version 3.x ou version ultérieure d'OpenSSL.

1. Générez une clé de chiffrement symétrique AES 256 bits

   Cette commande génère une clé de chiffrement symétrique AES composée de 256 bits aléatoires et l'enregistre dans le fichier `aes-key.bin`

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. Chiffrez votre élément de clé avec la clé de chiffrement symétrique AES

   Cette commande chiffre l'élément de votre clé avec la clé de chiffrement symétrique AES et enregistre le contenu de la clé chiffrée dans le fichier `key-material-wrapped.bin`.

   Dans cet exemple de commande :

   - *`PlaintextKeyMaterial.bin`* est le fichier qui contient l'élément de clé que vous importez, tel que `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der`, ou `ECC_NIST_P521_PrivateKey.der`.
   - *`aes-key.bin`* est le fichier qui contient la clé de chiffrement symétrique AES 256 bits que vous avez générée dans la commande précédente.

   ```
   # Encrypt your key material with the AES symmetric encryption key
   $ openssl enc -id-aes256-wrap-pad \
           -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
           -iv A65959A6 \
           -in PlaintextKeyMaterial.bin\
           -out key-material-wrapped.bin
   ```

3. Chiffrez votre clé de chiffrement symétrique AES avec la clé publique

   Cette commande chiffre votre clé de chiffrement symétrique AES avec la clé publique que vous avez téléchargée et l'algorithme d'encapsulage RSAES_OAEP_SHA_256, la code DER et l'enregistre dans le fichier `aes-key-wrapped.bin`.

   Dans cet exemple de commande :

- *WrappingPublicKey.bin* est le fichier qui contient la clé publique d'encapsulage téléchargée. Si vous avez téléchargé la clé publique à partir de la console, ce fichier est nommé `wrappingKey_`*KMS key_key_ID_timestamp* (par exemple, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`)
- *aes-key.bin* est le fichier qui contient la clé de chiffrement symétrique AES 256 bits que vous avez générée dans la première commande de cette séquence d'exemple.

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
    -encrypt \
    -in aes-key.bin \
    -out aes-key-wrapped.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha256 \
    -pkeyopt rsa_mgf1_md:sha256
```

4. Générez le fichier à importer

   Concaténez le fichier contenant l'élément de clé chiffré et le fichier contenant la clé AES chiffrée. Enregistrez-les dans le fichier `EncryptedKeyMaterial.bin`, qui est le fichier que vous allez importer dans le Étape 4 : Importation des éléments de clé.

   Dans cet exemple de commande :

   - *key-material-wrapped.bin* est le fichier qui contient votre élément de clé chiffré.
   - *aes-key-wrapped.bin* est le fichier qui contient la clé de chiffrement AES chiffrée.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

Passez à Étape 4 : Importation des éléments de clé.

Importation des éléments de clé – Étape 4 : Importation des éléments de clé

Après avoir chiffré vos éléments de clé, vous pouvez importer les éléments de clé à utiliser avec une AWS KMS key. Pour importer les éléments de clé, vous devez télécharger les éléments de clé chiffrés à partir de l'Étape 3 : Chiffrement des éléments de clé et le jeton d'importation que vous avez téléchargé à l'Étape 2 : Téléchargement de la clé publique d'encapsulage et du jeton d'importation. Vous devez importer les éléments de clé dans la même clé KMS que vous avez spécifiée lorsque vous avez téléchargé la clé publique et le jeton d'importation. Lorsque l'élément de clé est importé avec succès, l'état de la clé KMS passe à `Enabled`, et vous pouvez utiliser la clé KMS dans des opérations de chiffrement.

Lorsque vous importez les éléments de clé, vous pouvez éventuellement définir une date d'expiration pour ceux-ci. Lorsque les éléments de clé expirent, AWS KMS supprime les éléments de clé et la clé KMS devient inutilisable. Pour utiliser la clé KMS dans des opérations de chiffrement, vous devez réimporter le même élément de clé. Après avoir importé vos éléments de clé, vous ne pouvez pas définir, modifier ou annuler la date d'expiration de l'importation en cours. Pour modifier ces valeurs, vous devez supprimer et réimporter les mêmes éléments de clé.

Pour importer du matériel clé, vous pouvez utiliser la AWS KMS console ou l'ImportKeyMaterialAPI. Vous pouvez utiliser l'API directement en exécutant des requêtes HTTP, ou en utilisant un kit SDK AWS, l'AWS Command Line Interface ou AWS Tools for PowerShell.

Lorsque vous importez le matériel clé, une ImportKeyMaterialentrée est ajoutée à votre AWS CloudTrail journal pour enregistrer l'`ImportKeyMaterial`opération. L' CloudTrail entrée est la même que vous utilisiez la AWS KMS console ou l'AWS KMSAPI.

Définir un délai d'expiration (facultatif)

Lorsque vous importez les éléments de clé de votre clé KMS, vous pouvez définir une date et une heure d'expiration facultatives pour les éléments de clé pouvant aller jusqu'à 365 jours à compter de la date d'importation. Lorsque les éléments de clé importés expirent, AWS KMS les supprime. Cette action change l'état de la clé KMS en `PendingImport`, ce qui l'empêche d'être utilisée dans toute opération cryptographique. Pour utiliser la clé KMS, vous devez réimporter une copie des éléments de clé originaux.

S'assurer que les éléments de clé importés expirent fréquemment peut vous aider à satisfaire aux exigences réglementaires, mais cela introduit un risque supplémentaire pour les données chiffrées au moyen de la clé KMS. Tant que vous n'avez pas réimporté une copie des éléments de clé originaux, une clé KMS dont les éléments de clé ont expiré est inutilisable, et toutes les données chiffrées au

moyen de la clé KMS sont inaccessibles. Si vous ne parvenez pas à réimporter les éléments de clé pour quelque raison que ce soit, y compris la perte de votre copie des éléments de clé originaux, la clé KMS est définitivement inutilisable et les données chiffrées au moyen de la clé KMS sont irrécupérables.

Pour atténuer ce risque, assurez-vous que votre copie des éléments de clé importés est accessible et concevez un système pour supprimer et réimporter les éléments de clé avant qu'ils n'expirent et n'interrompent votre charge de travail AWS. Nous vous recommandons de programmer une alerte pour l'expiration de vos éléments de clé importés, afin de disposer de suffisamment de temps pour réimporter les éléments de clé avant leur expiration. Vous pouvez également utiliser vos CloudTrail journaux pour auditer les opérations d'importation (et de réimportation) de matériel clé et de suppression de matériel clé importé, ainsi que l'AWS KMSopération de suppression de matériel clé expiré.

Vous ne pouvez pas importer des éléments de clé différents dans la clé KMS, et AWS KMS ne peut pas restaurer, récupérer ou reproduire les éléments de clé supprimés. Au lieu de définir une date d'expiration, vous pouvez supprimer et réimporter périodiquement et par programmation les éléments de clé importés, mais les exigences relatives à la conservation d'une copie des éléments de clé originaux sont les mêmes.

Vous déterminez si et quand les éléments de clé importés expirent lorsque vous importez les éléments de clé importés. Mais vous pouvez activer et désactiver l'expiration, ou définir un nouveau délai d'expiration en supprimant et en réimportant les éléments de clé. Utilisez le `ExpirationModel` paramètre de ImportKeyMaterialpour activer et désactiver l'expiration (`KEY_MATERIAL_DOES_NOT_EXPIRE`) et le `ValidTo` paramètre pour définir le délai d'expiration. `KEY_MATERIAL_EXPIRES` Le délai maximal est de 365 jours à compter de la date d'importation ; il n'y a pas de minimum, mais le délai doit être dans le futur.

Importer les éléments de clé (console)

Vous pouvez utiliser AWS Management Console pour importer les éléments de clé.

1. Si vous êtes sur la page Téléchargez votre élément de clé encapsulé, passez à Step 8.
2. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
3. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
4. Dans le volet de navigation, choisissez Clés gérées par le client.

5. Choisissez l'ID de clé ou l'alias de la clé KMS pour laquelle vous avez téléchargé la clé publique et le jeton d'importation.
6. Choisissez l'onglet Cryptographic configuration (Configuration de chiffrement) et affichez ses valeurs. Les onglets se trouvent sur la page détaillée d'une clé KMS située sous la section General configuration (Configuration générale).

   Vous pouvez uniquement importer un élément de clé dans des clés KMS d'une Origine Externe (Importation d'éléments de clé). Pour plus d'informations sur la création de clés KMS avec des éléments de clé importés, veuillez consulter Importation de matériel clé pour les AWS KMS clés.
7. Choisissez l'onglet Key material (Éléments de clé) puis choisissez Import key material (Importation des éléments de clé). L'onglet Key material (Éléments de clés) apparaît uniquement pour les clés KMS dont la valeur Origin (Origine) est définie sur (Externe (Importation d'éléments de clé).

   Si vous avez téléchargé l'élément de clé, le jeton d'importation et chiffré l'élément de clé, choisissez Next (Suivant).
8. Dans la section Éléments clés chiffrés et jeton d'importation, procédez comme suit.

   a. Sous Élément de clé encapsulé, choisissez Choisir un fichier. Puis, chargez le fichier qui inclut vos clés encapsulées (chiffrées).

   b. Sous Jeton d'importation, choisissez Charger un fichier. Chargez le fichier qui inclut le jeton d'importation que vous avez téléchargé.
9. Sous Expiration option (Option d'expiration), déterminez si l'élément de clé expire. Pour définir la date et l'heure d'expiration, choisissez Date d'expiration des clés, et utilisez le calendrier pour sélectionner une date et une heure. Vous pouvez spécifier une date jusqu'à 365 jours à compter de la date et de l'heure actuelles.
10. Choisissez Charger une clé.

Importation des éléments de clé (API AWS KMS)

Pour importer du matériel clé, utilisez l'ImportKeyMaterialopération. Les exemples suivants utilisent l'AWS CLI, mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Pour utiliser cet exemple :

1. Remplacer *1234abcd-12ab-34cd-56ef-1234567890ab* par l'ID de la clé KMS que vous avez spécifié lorsque vous avez téléchargé la clé publique et le jeton d'importation. Pour identifier la clé

KMS, utilisez son ID de clé ou son ARN de clé. Vous ne pouvez pas utiliser un nom d'alias ou un ARN d'alias pour cette opération.

2. Remplacez *EncryptedKeyMaterial.bin* par le nom du fichier qui contient les clés chiffrées.
3. Remplacez *ImportToken.bin* par le nom du fichier qui contient le jeton d'importation.
4. Si vous souhaitez que l'élément de clé importé expire, définissez la valeur du paramètre `expiration-model` sur sa valeur par défaut, `KEY_MATERIAL_EXPIRES`, ou omettez le paramètre `expiration-model`. Procédez ensuite au remplacement de la valeur du paramètre `valid-to` par la date et l'heure auxquelles vous souhaitez que l'élément de clé expire. La date et l'heure peuvent aller jusqu'à 365 jours à compter de la date de la requête.

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2023-06-17T12:00:00-08:00
```

Si vous souhaitez que l'élément de clé importé n'expire pas, définissez la valeur du paramètre `expiration-model` sur `KEY_MATERIAL_DOES_NOT_EXPIRE`, et omettez le paramètre `valid-to` de la commande.

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_DOES_NOT_EXPIRE
```

> **Tip**
>
> Si la commande échoue, vous pouvez voir un `KMSInvalidStateException` ou un `NotFoundException`. Vous pouvez réessayer la demande.

Magasins de clés personnalisés

Un magasin de clés est un emplacement sécurisé pour stocker les clés cryptographiques. Le magasin de clés par défaut de AWS KMS prend également en charge les méthodes pour générer et gérer les clés qu'il stocke. Par défaut, les éléments de clé cryptographiques pour l'AWS KMS

keys que vous créez dans AWS KMS sont générés dans et protégés par des modules de sécurité matérielle (HSM) qui sont des modules cryptographiques validés FIPS 140-2. Les éléments de clé pour vos clés KMS ne laissent jamais les HSM non chiffrés.

Cependant, si vous avez besoin d'un contrôle encore plus poussé des modules HSM, vous pouvez créer un magasin de clés personnalisé.

Un magasin de clés personnalisé est un magasin de clés logique dans AWS KMS qui est soutenu par un gestionnaire de clés en dehors d'AWS KMS que vous possédez et gérez. Les magasins de clés personnalisés combinent l'interface de gestion des clés pratique et complète d'AWS KMS avec la possibilité de posséder et de contrôler les éléments de clé et les opérations cryptographiques. Lorsque vous utilisez une clé KMS dans un magasin de clés personnalisé, les opérations cryptographiques sont effectuées par votre gestionnaire de clés en utilisant vos clés cryptographiques. Par conséquent, vous endossez davantage de responsabilités en ce qui concerne la disponibilité et la durabilité des clés cryptographiques, ainsi que le fonctionnement des HSM.

AWS KMS prend en charge deux types de magasins de clés personnalisés.

- Un magasin de clés AWS CloudHSM est un magasin de clés personnalisé AWS KMS soutenu par un cluster AWS CloudHSM. Lorsque vous créez une clé KMS dans votre magasin de clés AWS CloudHSM, AWS KMS génère une clé symétrique Advanced Encryption Standard (AES) de 256 bits, persistante et non exportable dans le cluster AWS CloudHSM associé. Ces éléments de clé ne quittent jamais vos clusters AWS CloudHSM sans être chiffrés. Lorsque vous utilisez une clé KMS dans le magasin de clés AWS CloudHSM, les opérations cryptographiques sont effectuées dans les modules HSM du cluster. Les clusters AWS CloudHSM sont soutenus par des modules de sécurité matérielle (HSM) certifiés FIPS 140-2 niveau 3.
- Un magasin de clés externe est un magasin de clés personnalisé AWS KMS soutenu par un gestionnaire de clés externe en dehors d'AWS que vous possédez et contrôlez. Lorsque vous utilisez une clé KMS dans votre magasin de clés externe, toutes les opérations de chiffrement et de déchiffrement sont effectuées par votre gestionnaire de clés externe à l'aide de vos clés cryptographiques. Les magasins de clés externes sont conçus pour prendre en charge divers gestionnaires de clés externes provenant de différents fournisseurs.

  AWS KMS ne voit, n'accède et n'interagit jamais directement avec votre gestionnaire de clés externe ou vos clés cryptographiques. Lorsque vous chiffrez ou déchiffrez à l'aide d'une clé KMS dans un magasin de clés externe, l'opération est effectuée par votre gestionnaire de clés externe à l'aide de vos clés externes. Vous conservez le contrôle total de vos clés cryptographiques, y compris la possibilité de refuser ou d'arrêter une opération cryptographique sans interagir avec

AWS. Cependant, en raison de la distance et du traitement supplémentaire, les clés KMS dans un magasin de clés externe pourraient présenter une latence et des performances moindres, de même que des caractéristiques de disponibilité différentes de celles des clés KMS avec des éléments de clé dans AWS KMS. Pour plus d'informations sur les gestionnaires de clés compatibles avec la fonctionnalité de stockage de clés externes AWS KMS, consultez Quels fournisseurs externes prennent en charge la spécification XKS Proxy ? dans les AWS Key Management ServiceFAQ.

Ces deux types de magasins de clés personnalisés sont très différents du magasin de clés AWS KMS standard et l'un de l'autre. Leurs modèles de sécurité, leur lieu de responsabilité, leurs performances, leur prix et leurs cas d'utilisation sont également très différents. Avant de choisir un magasin de clés personnalisé, lisez la documentation associée et confirmez que les responsabilités supplémentaires en matière de configuration et de maintenance constituent un compromis judicieux pour le supplément de contrôle. Toutefois, si les règles et règlements dans le cadre desquels vous opérez exigent un contrôle direct des éléments de clé, un magasin de clés personnalisé pourrait constituer un bon choix pour vous.

Fonctions non prises en charge

AWS KMS ne prend pas en charge les fonctions suivantes dans les magasins de clés personnalisés.

- Clés KMS asymétriques
- Paires de clés de données asymétriques
- Clés KMS HMAC
- Clés KMS avec des éléments de clé importés
- Rotation automatique des clés
- Clés multi-région

Rubriques

- AWS CloudHSM magasins clés
- Magasins de clés externes

AWS CloudHSM magasins clés

Un magasin de AWS CloudHSM clés est un magasin de clés personnalisé soutenu par un AWS CloudHSM cluster. Lorsque vous créez un magasin AWS KMS keyde clés personnalisé, vous AWS KMS générez et stockez des éléments clés non extractibles pour la clé KMS dans un AWS CloudHSM cluster que vous possédez et gérez. Lorsque vous utilisez une clé KMS dans un magasin de clés personnalisé, les opérations de chiffrement sont effectuées dans les modules HSM du cluster. Cette fonctionnalité combine la commodité et AWS KMS l'intégration généralisée d'un AWS CloudHSM cluster dans votre Compte AWS.

AWS KMS fournit un support complet de console et d'API pour la création, l'utilisation et la gestion de vos magasins de clés personnalisés. Vous pouvez utiliser les clés KMS dans votre magasin de clés personnalisé de la même manière que vous utilisez n'importe quelle clé KMS. Par exemple, vous pouvez utiliser les clés KMS pour générer des clés de données et chiffrer des données. Vous pouvez également utiliser les clés KMS dans votre magasin de clés personnalisé avec des AWS services prenant en charge les clés gérées par le client.

Est-ce que j'ai besoin d'un magasin de clés personnalisé ?

Pour la plupart des utilisateurs, le magasin de AWS KMS clés par défaut, qui est protégé par des modules cryptographiques validés par la norme FIPS 140-2, répond à leurs exigences de sécurité. Il n'est pas nécessaire d'ajouter une couche supplémentaire de responsabilité de maintenance ou une dépendance à l'égard d'un service supplémentaire.

Cependant, vous pouvez envisager la création d'un magasin de clés personnalisé si votre organisation possède l'une des exigences suivantes :

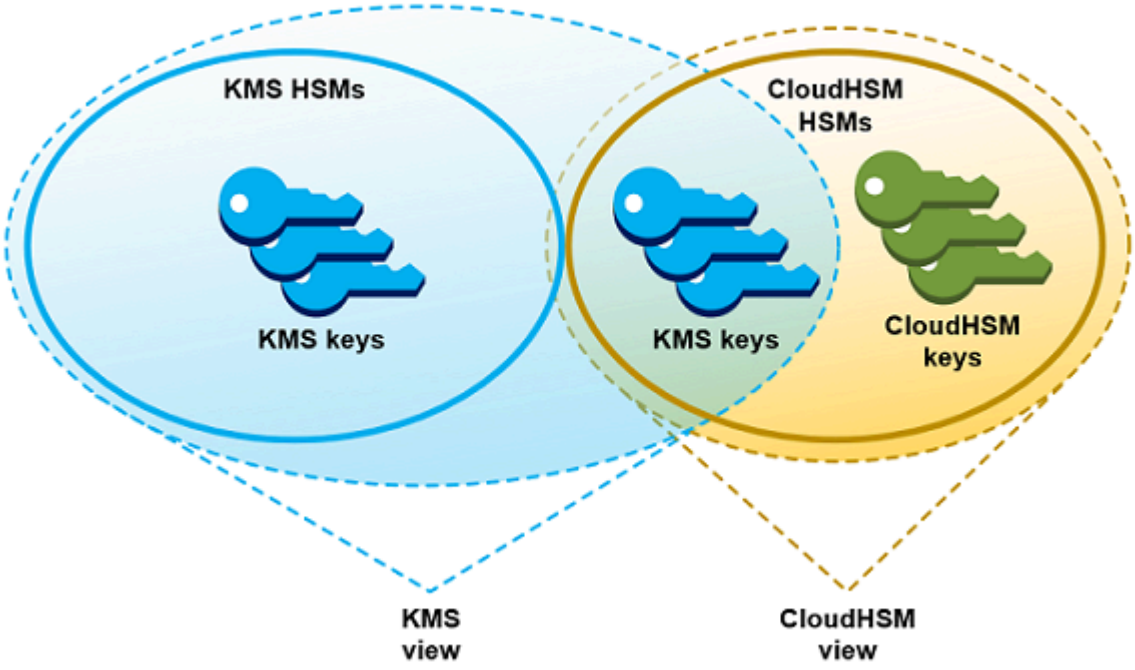
- Vous avez des clés qui doivent explicitement être protégées dans un HSM à locataire unique ou dans un HSM sur lequel vous avez un contrôle direct.
- Vous devez être en mesure de retirer immédiatement les éléments clés de AWS KMS.
- Vous devez être en mesure d'auditer toute utilisation de vos clés indépendamment de AWS KMS ou AWS CloudTrail.

Comment fonctionnent les magasins de clés personnalisés ?

Chaque magasin de clés personnalisé est associé à un AWS CloudHSM cluster dans votre Compte AWS. Lorsque vous connectez le magasin de clés personnalisé à son cluster, il AWS KMS crée l'infrastructure réseau pour prendre en charge la connexion. Il se connecte ensuite

au AWS CloudHSM client clé du cluster à l'aide des informations d'identification d'un utilisateur cryptographique dédié du cluster.

Vous créez et gérez vos magasins de clés personnalisés dans AWS KMS et vous créez et gérez vos clusters HSM dans AWS CloudHSM. Lorsque vous créez AWS KMS keys dans un magasin de clés AWS KMS personnalisé, vous visualisez et gérez les clés KMS dans AWS KMS. Mais vous pouvez également afficher et gérer leurs éléments clés dans AWS CloudHSM, comme vous le feriez pour les autres clés du cluster.



Vous pouvez créer des clés KMS de chiffrement symétriques à partir du contenu clé généré par AWS KMS votre magasin de clés personnalisé. Utilisez ensuite les mêmes techniques pour afficher et gérer les clés KMS dans votre magasin de clés personnalisé que celles que vous utilisez pour les clés KMS dans le magasin de AWS KMS clés. Vous pouvez contrôler l'accès aux politiques IAM et aux politiques de clé, créer des balises et des alias, activer et désactiver les clés KMS, et planifier la suppression de clés. Vous pouvez utiliser les clés KMS pour des opérations cryptographiques et les utiliser avec AWS des services intégrés à AWS KMS.

En outre, vous avez un contrôle total sur le AWS CloudHSM cluster, notamment en créant et en supprimant des HSM et en gérant les sauvegardes. Vous pouvez utiliser le AWS CloudHSM client et les bibliothèques logicielles prises en charge pour visualiser, auditer et gérer les éléments clés de vos clés KMS. Lorsque le magasin de clés personnalisé est déconnecté, il AWS KMS ne peut pas y accéder et les utilisateurs ne peuvent pas utiliser les clés KMS du magasin de clés personnalisé

pour des opérations cryptographiques. Cette nouvelle couche de contrôle fait du magasin de clés personnalisé une solution puissante pour les organisations qui en ont besoin.

Où commencer ?

Pour créer et gérer un magasin de AWS CloudHSM clés, vous utilisez les fonctionnalités de AWS KMS et AWS CloudHSM.

1. Commencez par AWS CloudHSMCréez un cluster AWS CloudHSM actif ou sélectionnez un cluster existant. Le cluster doit avoir au moins deux modules HSM actifs de différentes zones de disponibilité. Ensuite, créez un compte CU (utilisateur de chiffrement) dédié dans ce cluster pour AWS KMS.
2. Dans AWS KMS, créez un magasin de clés personnalisé associé au AWS CloudHSM cluster sélectionné. AWS KMS fournit une interface de gestion complète qui vous permet de créer, d'afficher, de modifier et de supprimer vos banques de clés personnalisées.
3. Lorsque vous êtes prêt à utiliser votre magasin de clés personnalisé, connectez-le au AWS CloudHSM cluster associé. AWS KMS crée l'infrastructure réseau dont elle a besoin pour prendre en charge la connexion. Ensuite, il se connecte au cluster à l'aide des informations d'identification du compte CU dédié afin de générer et de gérer les clés dans le cluster.
4. Vous pouvez désormais créer des clés KMS de chiffrement symétriques dans votre magasin de clés personnalisé. Il vous suffit de spécifier le magasin de clés personnalisé lorsque vous créez la clé KMS.

Si vous vous retrouvez bloqué à un moment, vous pouvez obtenir de l'aide dans la rubrique Dépannage d'un magasin de clés personnalisé. Si vous ne trouvez pas de réponse à votre question, utilisez le lien situé en bas de chaque page de ce guide ou publiez une question sur le AWS Key Management Service forum de discussion.

Quotas

AWS KMS autorise jusqu'à 10 magasins de clés personnalisés dans chaque Compte AWS région, y compris les magasins de AWS CloudHSM clés et les magasins de clés externes, quel que soit leur état de connexion. En outre, il existe des quotas de AWS KMS demandes concernant l'utilisation des clés KMS dans un magasin de AWS CloudHSM clés.

Tarification

Pour plus d'informations sur le coût des magasins de clés AWS KMS personnalisés et des clés gérées par le client dans un magasin de clés personnalisé, consultez AWS Key Management Service

les tarifs. Pour plus d'informations sur le coût des AWS CloudHSM clusters et des HSM, consultez la section AWS CloudHSM Tarification.

Régions

AWS KMS prend en charge les AWS CloudHSM principaux magasins dans Régions AWS tous AWS KMS les pays concernés, à l'exception de l'Asie-Pacifique (Melbourne), de la Chine (Pékin), de la Chine (Ningxia) et de l'Europe (Espagne).

Fonctions non prises en charge

AWS KMS ne prend pas en charge les fonctionnalités suivantes dans les magasins de clés personnalisés.

- Clés KMS asymétriques
- Paires de clés de données asymétriques
- Clés KMS HMAC
- Clés KMS avec des éléments de clé importés
- Rotation automatique des clés
- Clés multi-région

Rubriques

- Concepts de magasins de clés AWS CloudHSM
- Contrôler l'accès à votre magasin de clés AWS CloudHSM
- Gérer un magasin de clés personnalisé CloudHSM
- Gérer les clés KMS dans un magasin de clés CloudHSM
- Dépannage d'un magasin de clés personnalisé

Concepts de magasins de clés AWS CloudHSM

Cette rubrique explique certains des concepts utilisés dans les magasins de clés AWS CloudHSM.

Magasin de clés AWS CloudHSM

Un magasin de clés AWS CloudHSM est un magasin de clés personnalisé qui est associé à un cluster AWS CloudHSM que vous possédez et gérez. Les clusters AWS CloudHSM sont soutenus par des modules de sécurité matérielle (HSM) certifiés selon la norme FIPS 140-2 niveau 3.

Lorsque vous créez une clé KMS dans votre magasin de clés AWS CloudHSM, AWS KMS génère une clé symétrique Advanced Encryption Standard (AES) de 256 bits, persistante et non exportable dans le cluster AWS CloudHSM associé. Cette clé ne quitte jamais vos modules HSM non chiffrés. Lorsque vous utilisez une clé KMS dans un magasin de clés AWS CloudHSM, les opérations cryptographiques sont effectuées dans les modules HSM du cluster.

Les magasins de clés AWS CloudHSM combinent l'interface de gestion de clés pratique et complète d'AWS KMS avec les contrôles supplémentaires fournis par un cluster AWS CloudHSM de votre Compte AWS. Cette fonction intégrée vous permet de créer, gérer et utiliser des clés KMS dans AWS KMS tout en conservant une maîtrise totale des modules HSM qui stockent leurs éléments de clé, y compris la gestion des clusters, les modules HSM et les sauvegardes. Vous pouvez utiliser la console et les API AWS KMS pour gérer le magasin de clés AWS CloudHSM et ses clés KMS. Vous pouvez également utiliser la console AWS CloudHSM, les API, les logiciels client et les bibliothèques logicielles connexes pour gérer le cluster associé.

Vous pouvez afficher et gérer votre magasin de clés AWS CloudHSM, modifier ses propriétés et le connecter et le déconnecter de son cluster AWS CloudHSM associé. Si vous devez supprimer un magasin de clés AWS CloudHSM, vous devez d'abord supprimer les clés KMS dans le magasin de clés AWS CloudHSM en planifiant leur suppression et en attendant jusqu'à ce que la période de grâce expire. La suppression du magasin de clés AWS CloudHSM supprime la ressource d'AWS KMS, mais elle n'a pas d'incidence sur votre cluster AWS CloudHSM.

AWS CloudHSMCluster

Chaque magasin de clés AWS CloudHSM est associé à exactement un cluster AWS CloudHSM. Lorsque vous créez une AWS KMS key dans votre magasin de clés AWS CloudHSM, AWS KMS crée ses éléments de clé dans le cluster associé. Lorsque vous utilisez une clé KMS dans votre magasin de clés AWS CloudHSM, les opérations cryptographiques sont effectuées dans le cluster associé.

Chaque cluster AWS CloudHSM ne peut être associé qu'à un seul magasin de clés AWS CloudHSM. Le cluster que vous choisissez ne peut pas être associé à un autre magasin de clés AWS CloudHSM ou partager un historique de sauvegarde avec un cluster qui est associé à un autre magasin de clés AWS CloudHSM. Le cluster doit être initialisé et actif, et doit être dans les mêmes Compte AWS et région que le magasin de clés AWS CloudHSM. Vous pouvez créer un nouveau cluster ou en utiliser un existant. AWS KMS ne nécessite pas un accès exclusif au cluster. Pour créer des clés KMS dans le magasin de clés AWS CloudHSM, son cluster associé doit contenir au moins deux modules HSM actifs. Toutes les autres opérations nécessitent un seul HSM.

Vous spécifiez le cluster AWS CloudHSM lorsque vous créez le magasin de clés AWS CloudHSM et vous ne pouvez pas le modifier. Cependant, vous pouvez remplacer n'importe quel cluster qui partage un historique de sauvegardes avec le cluster d'origine. Cela vous permet de supprimer le cluster, si nécessaire, et de le remplacer-le par un cluster créé à partir de l'une de ses sauvegardes. Vous gardez une maîtrise totale du cluster AWS CloudHSM associé, ce qui vous permet de gérer des utilisateurs et des clés, de créer et de supprimer des modules HSM, et d'utiliser et de gérer les sauvegardes.

Lorsque vous êtes prêt à utiliser votre magasin de clés AWS CloudHSM, vous pouvez le connecter à son cluster AWS CloudHSM associé. Vous pouvez connecter et déconnecter votre magasin de clés personnalisé à tout moment. Lorsqu'un magasin de clés personnalisé est connecté, vous pouvez créer et utiliser ses clés KMS. Lorsqu'il est déconnecté, vous pouvez afficher et gérer le magasin de clés AWS CloudHSM et ses clés KMS. Toutefois, vous ne pouvez pas créer de nouvelles clés KMS ou utiliser les clés KMS du magasin de clés AWS CloudHSM pour les opérations cryptographiques.

Utilisateur de chiffrement **`kmsuser`**

Pour créer et gérer les éléments de clé dans le cluster AWS CloudHSM associé en votre nom, AWS KMS utilise un utilisateur de chiffrement (CU) AWS CloudHSM dédié dans le cluster nommé `kmsuser`. Le `kmsuser` CU est un compte CU standard qui est automatiquement synchronisé à tous les HSM du cluster et qui est enregistré dans les sauvegardes de clusters.

Avant de créer votre magasin de clés AWS CloudHSM, vous créez un compte CU `kmsuser` dans votre cluster AWS CloudHSM à l'aide de la commande createUser dans cloudhsm_mgmt_util. Ensuite, lorsque vous créez le magasin de clés AWS CloudHSM, vous fournissez le mot de passe du compte `kmsuser` à AWS KMS. Lorsque vous connectez le magasin de clés personnalisé, AWS KMS se connecte au cluster en tant qu'utilisateur de chiffrement (CU) `kmsuser` et effectue une rotation de ses mots de passe. AWS KMS chiffre votre mot de passe `kmsuser` avant de le stocker en sécurité. Lorsque le mot de passe a effectué une rotation, le nouveau mot de passe est chiffré et stocké de la même manière.

AWS KMS reste connecté en tant que `kmsuser` jusqu'à ce que le magasin de clés AWS CloudHSM soit connecté. Vous ne devez pas utiliser ce compte CU à d'autres fins. Toutefois, vous gardez le contrôle ultime du compte CU `kmsuser`. A tout moment, vous pouvez rechercher les descripteurs de clés des clés que `kmsuser` détient. Si nécessaire, vous pouvez déconnecter le magasin de clés personnalisé, modifier le mot de passe `kmsuser`, vous connecter au cluster en tant que `kmsuser` et afficher et gérer les clés que `kmsuser` détient.

Pour obtenir des instructions sur la création de votre compte CU `kmsuser`, consultez Créer l'utilisateur de chiffrement (CU) `kmsuser`.

Clés KMS dans un magasin de clés AWS CloudHSM

Vous pouvez utiliser AWS KMS ou l'API AWS KMS pour créer des AWS KMS keys dans un magasin de clés AWS CloudHSM. Vous utilisez la même technique que celle que vous utiliseriez sur n'importe quelle clé KMS. La seule différence est que vous devez identifier le magasin de clés AWS CloudHSM et spécifier que l'origine des éléments de clé est le cluster AWS CloudHSM.

Lorsque vous créez une clé KMS dans un magasin de clés AWS CloudHSM, AWS KMS crée la clé KMS dans AWS KMS et génère des éléments de clé symétriques Advanced Encryption Standard (AES) de 256 bits, persistants, non exportables dans leur cluster associé. Lorsque vous utilisez la clé AWS KMS dans une opération cryptographique, l'opération est effectuée dans leAWS CloudHSMcluster utilisant la clé AES basée sur un cluster. Bien que AWS CloudHSM prenne en charge les clés symétriques et asymétriques de différents types, les magasins de clés AWS CloudHSM ne prennent en charge que les clés de chiffrement AES symétriques.

Vous pouvez afficher les clés KMS d'un magasin de clés AWS CloudHSM dans la console AWS KMS et utiliser les options de la console pour afficher l'ID de magasin de clés personnalisé. Vous pouvez également utiliser cette DescribeKeyopération pour trouver l'ID du magasin de AWS CloudHSM clés et l'ID AWS CloudHSM du cluster.

Les clés KMS d'un magasin de clés AWS CloudHSM fonctionnent comme n'importe quelle clé KMS dans AWS KMS. Les utilisateurs autorisés ont besoin des mêmes autorisations pour utiliser et gérer les clés KMS. Vous utilisez les mêmes procédures et opérations d'API de la console pour afficher et gérer les clés KMS d'un magasin de clés AWS CloudHSM. Cela inclut l'activation et la désactivation de clés KMS, la création et l'utilisation de balises et d'alias, et la définition et la modification des politiques de clé et des politiques IAM. Vous pouvez utiliser les clés KMS dans un magasin de clés AWS CloudHSM pour les opérations cryptographiques, et les utiliser avec des services AWS intégrés qui prennent en charge l'utilisation de clés gérées par le client. Cependant, vous ne pouvez pas activer la rotation automatique des clés ou importer des éléments de clé vers une clé KMS dans un magasin de clés AWS CloudHSM.

Vous pouvez également utiliser la même procédure pour planifier la suppression d'une clé KMS dans un magasin de clés AWS CloudHSM. Lorsque la période d'attente a expiré, AWS KMS supprime la clé KMS depuis KMS. Ensuite, il met tout en œuvre pour supprimer les éléments de clé de la clé KMS du cluster AWS CloudHSM associé. Cependant, il se peut que vous ayez besoin de supprimer manuellement les éléments de clé orphelins du cluster et de ses sauvegardes.

Contrôler l'accès à votre magasin de clés AWS CloudHSM

Vous utilisez les politiques IAM pour contrôler l'accès à votre magasin de clés AWS CloudHSM et votre cluster AWS CloudHSM. Vous pouvez utiliser les politiques IAM et les politiques de clé pour contrôler l'accès aux AWS KMS keys de votre magasin de clés AWS CloudHSM. Nous vous recommandons de fournir aux utilisateurs, groupes et rôles uniquement les autorisations dont ils ont besoin pour les tâches qu'ils sont susceptibles d'effectuer.

Rubriques

- Autoriser les gestionnaires et utilisateurs d'un magasin de clés AWS CloudHSM
- Autoriser AWS KMS à gérer AWS CloudHSM et les ressources Amazon EC2

Autoriser les gestionnaires et utilisateurs d'un magasin de clés AWS CloudHSM

Lors de la conception de votre magasin de clés AWS CloudHSM, veillez à ce que les principaux qui l'utilisent et le gèrent disposent uniquement des autorisations dont ils ont besoin. La liste suivante décrit les autorisations minimales requises pour les gestionnaires et les utilisateurs des magasins de clés AWS CloudHSM.

- Les principaux qui créent et gèrent votre magasin de clés AWS CloudHSM nécessitent l'autorisation suivante pour utiliser les opérations d'API du magasin de clés AWS CloudHSM.
  - `cloudhsm:DescribeClusters`
  - `kms:CreateCustomKeyStore`
  - `kms:ConnectCustomKeyStore`
  - `kms:DeleteCustomKeyStore`
  - `kms:DescribeCustomKeyStores`
  - `kms:DisconnectCustomKeyStore`
  - `kms:UpdateCustomKeyStore`
  - `iam:CreateServiceLinkedRole`
- Les principaux qui créent et gèrent le cluster AWS CloudHSM associé à votre magasin de clés AWS CloudHSM ont besoin d'une autorisation pour créer et initialiser un cluster AWS CloudHSM. Cela inclut l'autorisation de créer ou d'utiliser un cloud privé virtuel (VPC) Amazon, de créer des sous-réseaux et de créer une instance Amazon EC2. Elles peuvent également créer et supprimer des HSM, et gérer des sauvegardes. Pour obtenir la liste des autorisations requises, veuillez

consulter la rubrique Identity and access management for AWS CloudHSM (Gestion des identités et des accès pour) dans le Guide de l'utilisateur AWS CloudHSM.

- Les principaux qui créent et gèrent les AWS KMS keys de votre magasin de clés AWS CloudHSM nécessitent les mêmes autorisations que ceux qui créent et gèrent les clés KMS dans AWS KMS. La politique de clé par défaut pour une clé KMS d'un magasin de clés AWS CloudHSM est identique à la politique de clé par défaut pour les clés KMS dans AWS KMS. Le contrôle d'accès par attributs (ABAC), qui utilise des balises et des alias pour contrôler l'accès aux clés KMS, fonctionne également sur les clés KMS dans les magasins de clés AWS CloudHSM.
- Les principaux qui utilisent les clés KMS dans votre magasin de clés AWS CloudHSM pour les opérations cryptographiques ont besoin des autorisations pour effectuer les opérations cryptographiques avec la clé KMS, telles que kms:Decrypt. Vous pouvez fournir ces autorisations dans une politique de clé, ou une politique IAM. Cependant, les principaux n'ont pas besoin d'autorisations supplémentaires pour utiliser une clé KMS dans un magasin de clés AWS CloudHSM.

Autoriser AWS KMS à gérer AWS CloudHSM et les ressources Amazon EC2

Pour prendre en charge vos magasins de clés AWS CloudHSM, AWS KMS a besoin d'une autorisation pour obtenir des informations sur vos clusters AWS CloudHSM. Il a aussi besoin de l'autorisation de créer l'infrastructure réseau qui connecte votre magasin de clés AWS CloudHSM à son cluster AWS CloudHSM. Pour obtenir ces autorisations, AWS KMS crée le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStoreslié au service dans votre. Compte AWS Les utilisateurs qui créent des magasins de clés AWS CloudHSM doivent avoir l'autorisation `iam:CreateServiceLinkedRole` leur permettant de créer des rôles liés à un service.

Rubriques

- À propos du rôle lié à un service AWS KMS
- Création du rôle lié à un service
- Modifier la description du rôle lié à un service
- Supprimer le rôle lié à un service

À propos du rôle lié à un service AWS KMS

Un rôle lié à un service est un rôle IAM qui accorde à un service AWS l'autorisation d'appeler d'autres services AWS en votre nom. Il est conçu pour faciliter l'utilisation des fonctions de plusieurs services

AWS intégrés, sans avoir à créer et gérer des politiques IAM complexes. Pour plus d'informations, consultez Utilisation des rôles liés aux services pour AWS KMS.

Pour les magasins de AWS CloudHSM clés, AWS KMS crée le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStoreslié au service avec la AWSKeyManagementServiceCustomKeyStoresServiceRolePolicypolitique. Cette politique accorde au rôle les autorisations suivantes :

- cloudHSM:Describe* — détecte les modifications dans le AWS CloudHSM cluster attaché à votre magasin de clés personnalisé.
- ec2 : CreateSecurityGroup — utilisé lorsque vous connectez un magasin de AWS CloudHSM clés pour créer le groupe de sécurité qui permet le flux de trafic réseau entre AWS KMS et votre AWS CloudHSM cluster.
- ec2 : AuthorizeSecurityGroupIngress — utilisé lorsque vous connectez un magasin de AWS CloudHSM clés pour autoriser l'accès au réseau depuis AWS KMS le VPC qui contient AWS CloudHSM votre cluster.
- ec2 : CreateNetworkInterface — utilisé lorsque vous connectez un magasin de AWS CloudHSM clés pour créer l'interface réseau utilisée pour la communication entre AWS KMS et le AWS CloudHSM cluster.
- ec2 : RevokeSecurityGroupEgress — utilisé lorsque vous connectez un magasin de AWS CloudHSM clés pour supprimer toutes les règles sortantes du groupe de sécurité créé. AWS KMS
- ec2 : DeleteSecurityGroup — utilisé lorsque vous déconnectez un magasin de AWS CloudHSM clés pour supprimer les groupes de sécurité créés lors de la connexion du magasin de AWS CloudHSM clés.
- ec2 : DescribeSecurityGroups — utilisé pour surveiller les modifications apportées au groupe de sécurité AWS KMS créé dans le VPC contenant AWS CloudHSM votre cluster afin AWS KMS de fournir des messages d'erreur clairs en cas de défaillance.
- ec2 : DescribeVpcs — utilisé pour surveiller les modifications apportées au VPC qui contient AWS CloudHSM votre cluster afin AWS KMS de pouvoir fournir des messages d'erreur clairs en cas de défaillance.
- ec2 : DescribeNetworkAcls — utilisé pour surveiller les modifications des ACL du réseau pour le VPC qui contient votre AWS CloudHSM cluster afin de AWS KMS pouvoir fournir des messages d'erreur clairs en cas de défaillance.

- ec2 : DescribeNetworkInterfaces — utilisé pour surveiller les modifications des interfaces réseau AWS KMS créées dans le VPC qui contient AWS CloudHSM votre cluster afin AWS KMS de fournir des messages d'erreur clairs en cas de défaillance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudhsm:Describe*",
        "ec2:CreateNetworkInterface",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```

Étant donné que le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStoreslié au service n'est fiable que`cks.kms.amazonaws.com`, seul le rôle lié au service AWS KMS peut assumer ce rôle lié au service. Ce rôle est limité aux opérations dont AWS KMS a besoin pour afficher vos clusters AWS CloudHSM et connecter un magasin de clés AWS CloudHSM à son cluster AWS CloudHSM associé. Aucune autre autorisation n'est accordée à AWS KMS. Par exemple, AWS KMS n'a pas l'autorisation de créer, gérer ou supprimer vos clusters AWS CloudHSM, vos HSM ou vos sauvegardes.

Régions

À l'instar de la fonctionnalité AWS CloudHSM Key Stores, le AWSServiceRoleForKeyManagementServiceCustomKeyStoresrôle est pris en charge partout Régions AWS où il AWS KMS est disponible. AWS CloudHSM Pour obtenir la liste de Régions AWS pris en charge par chaque service, consultez les Points de terminaison et quotas AWS Key

Management Service et les points de terminaison et quotas AWS CloudHSM dans le Référence générale d'Amazon Web Services

Pour plus d'informations sur l'utilisation des rôles liés à un service par les services AWS, veuillez consulter la rubrique Utilisation des rôles liés à un service dans le Guide de l'utilisateur IAM.

Création du rôle lié à un service

AWS KMScrée automatiquement le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStoreslié au service dans votre magasin de clés Compte AWS lorsque vous créez un magasin de AWS CloudHSM clés, si le rôle n'existe pas déjà. Vous ne pouvez pas créer ou recréer directement ce rôle lié à un service.

Modifier la description du rôle lié à un service

Vous ne pouvez pas modifier le nom du rôle ou les déclarations de stratégie du rôle lié à un service AWSServiceRoleForKeyManagementServiceCustomKeyStores, mais vous pouvez modifier la description du rôle. Pour obtenir des instructions, veuillez consulter la rubrique Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Supprimer le rôle lié à un service

AWS KMSne supprime pas le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStoreslié au service, Compte AWS même si vous avez supprimé tous vos AWS CloudHSM principaux magasins. Bien qu'il n'existe actuellement aucune procédure permettant de supprimer le rôle AWSServiceRoleForKeyManagementServiceCustomKeyStoreslié à un service, AWS KMS elle n'assume pas ce rôle et n'utilise pas ses autorisations sauf si vous disposez de banques de AWS CloudHSM clés actives.

Gérer un magasin de clés personnalisé CloudHSM

À l'aide d'AWS Management Console de l'API AWS KMS, vous pouvez gérer un magasin de clés personnalisé. Par exemple, vous pouvez afficher un magasin de clés personnalisé, modifier ses propriétés, le connecter et le déconnecter de son cluster AWS CloudHSM associé et supprimer le magasin de clés personnalisé.

Rubriques

- Créer un magasin de clés AWS CloudHSM
- Afficher un magasin de clés AWS CloudHSM
- Modifier les paramètres d'un magasin de clés AWS CloudHSM

- Connecter et déconnecter un magasin de clés AWS CloudHSM
- Supprimer un magasin de clés AWS CloudHSM

Créer un magasin de clés AWS CloudHSM

Vous pouvez créer un ou plusieurs magasins de clés AWS CloudHSM dans votre compte. Chaque magasin de clés AWS CloudHSM est associé à un cluster AWS CloudHSM dans les mêmes Compte AWS et région. Avant de créer votre magasin de clés AWS CloudHSM, vous devez réunir les conditions préalables. Ensuite, avant de pouvoir utiliser votre magasin de clés AWS CloudHSM, vous devez le connecter à son cluster AWS CloudHSM.

> **Note**
>
> Si vous essayez de créer un magasin de clés AWS CloudHSM avec toutes les mêmes valeurs de propriétés qu'un magasin de clés AWS CloudHSM déconnecté existant, AWS KMS ne crée pas de magasin de clés AWS CloudHSM et ne lève pas d'exception ou d'erreur. Au lieu de cela, AWS KMS reconnaît que le doublon est la conséquence probable d'une nouvelle tentative et renvoie l'ID du magasin de clés AWS CloudHSM existant.

> **Tip**
>
> Vous n'avez pas besoin de connecter votre magasin de clés AWS CloudHSM immédiatement. Vous pouvez le conserver dans un état déconnecté jusqu'à ce que vous soyez prêt à l'utiliser. Cependant, afin de vérifier qu'il est correctement configuré, vous pouvez le connecter, afficher son état de connexion, puis le déconnecter.

Rubriques

- Rassembler les conditions requises
- Créer un magasin de clés AWS CloudHSM (console)
- Créer un magasin de clés AWS CloudHSM (API)

Rassembler les conditions requises

Chaque magasin de clés AWS CloudHSM est soutenu par un cluster AWS CloudHSM. Pour créer un magasin de clés AWS CloudHSM, vous devez spécifier un cluster AWS CloudHSM actif qui n'est pas

déjà associé à un autre magasin de clés. Vous devez également créer un utilisateur de chiffrement (CU) dédié dans les modules HSM du cluster que AWS KMS peut utiliser pour créer et gérer les clés en votre nom.

Avant de créer un magasin de clés AWS CloudHSM, procédez comme suit :

Sélectionner un cluster AWS CloudHSM

Chaque magasin de clés AWS CloudHSM est associé à exactement un cluster AWS CloudHSM. Lorsque vous créez une AWS KMS keys dans votre magasin de clés AWS CloudHSM, AWS KMS crée les métadonnées de clé KMS, telles qu'un ID et un Amazon Resource Name (ARN) dans AWS KMS. Il crée ensuite les éléments de clé dans les modules HSM du cluster associé. Vous pouvez créer un nouveau AWS CloudHSM cluster ou en utiliser un existant. AWS KMS ne nécessite pas un accès exclusif au cluster.

Le cluster AWS CloudHSM que vous sélectionnez est définitivement associé au magasin de clés AWS CloudHSM. Une fois que vous avez créé le magasin de clés AWS CloudHSM, vous pouvez modifier l'ID de cluster associé, mais le cluster que vous spécifiez doit partager un historique de sauvegarde avec le cluster d'origine. Pour utiliser un cluster non associé, vous devez créer un nouveau magasin de clés AWS CloudHSM.

Le cluster AWS CloudHSM que vous sélectionnez doit avoir les caractéristiques suivantes :

- Le cluster doit être actif.

  Vous devez créer le cluster, l'initialiser, installer le logiciel client AWS CloudHSM pour votre plateforme, puis activer le cluster. Pour plus d'informations, veuillez consulter la rubrique Getting started with AWS CloudHSM (Démarrer avec) dans le Guide de l'utilisateur AWS CloudHSM.
- Le cluster doit se trouver dans les mêmes compte et région que le magasin de clés AWS CloudHSM. Vous ne pouvez pas associer un magasin de clés AWS CloudHSM dans une région à un cluster dans une autre région. Pour créer une infrastructure de clés dans plusieurs régions, vous devez créez des magasins de clés AWS CloudHSM et des clusters dans chaque région.
- Le cluster ne peut pas être associé à un autre magasin de clés personnalisé à partir du même compte et de la même région. Chaque magasin de clés AWS CloudHSM dans le compte et la région doit être associé à un cluster AWS CloudHSM différent. Vous ne pouvez pas spécifier un cluster qui est déjà associé à un magasin de clés personnalisé ou un cluster qui partage un historique des sauvegardes avec un cluster associé. Les clusters qui partagent un historique

des sauvegardes historique ont le même certificat de cluster. Pour afficher le certificat de cluster d'un cluster, utilisez la AWS CloudHSM console ou l'DescribeClustersopération.

Si vous sauvegardez un cluster AWS CloudHSM dans une autre région, il est considéré comme un cluster différent et vous pouvez associer la sauvegarde à un magasin de clés personnalisé dans sa région. Toutefois, les clés KMS des deux magasins de clés personnalisés ne sont pas interopérables, même si elles possèdent la même clé de sauvegarde. AWS KMS lie les métadonnées au texte chiffré afin qu'il ne puisse être déchiffré que par la clé KMS qui l'a chiffré.

- Le cluster doit être configuré avec des sous-réseaux privés dans au moins deux zones de disponibilité de la région. Comme AWS CloudHSM n'est pas pris en charge dans toutes les zones de disponibilité, nous vous recommandons de créer des sous-réseaux privés dans toutes les zones de disponibilité de la région. Vous ne pouvez pas reconfigurer les sous-réseaux d'un cluster existant, mais vous pouvez créer un cluster à partir d'une sauvegarde avec différents sous-réseaux dans la configuration du cluster.

  > **Important**
  >
  > Après avoir créé votre magasin de clés AWS CloudHSM, ne supprimez aucun des sous-réseaux privés configurés pour son cluster AWS CloudHSM. Si AWS KMS ne trouve pas tous les sous-réseaux dans la configuration du cluster, les tentatives de connexion au magasin de clés personnalisé échouent avec un état d'erreur de connexion `SUBNET_NOT_FOUND`. Pour plus de détails, veuillez consulter Comment corriger un échec de connexion.

- Le groupe de sécurité du cluster (`cloudhsm-cluster-<cluster-id>-sg`) doit inclure des règles entrantes et sortantes qui autorisent le trafic TCP sur les ports 2223-2225. La source des règles entrantes et la destination des règles sortantes doit correspondre à l'ID du groupe de sécurité. Ces règles sont définies par défaut lorsque vous créez le cluster. Ne pas les supprimer ou les modifier.
- Le cluster doit avoir au moins deux modules HSM actifs dans différentes zones de disponibilité. Pour vérifier le nombre de HSM, utilisez la AWS CloudHSM console ou l'DescribeClustersopération. Si nécessaire, vous pouvez ajouter un module HSM.

Recherche le certificat approuvé (ou « trust anchor »)

Lorsque vous créez un magasin de clés personnalisé, vous devez charger le certificat de point de confiance pour le cluster AWS CloudHSM dans AWS KMS. AWS KMS a besoin du certificat

de point de confiance pour connecter le magasin de clés AWS CloudHSM à son cluster AWS CloudHSM associé.

Chaque cluster AWS CloudHSM actif a un certificat approuvé (ou« trust anchor »). Lorsque vous initialisez le cluster, vous devez générer ce certificat, l'enregistrer dans le fichier `customerCA.crt` et le copier sur les hôtes qui se connectent au cluster.

Créez l'**`kmsuser`** Utilisateur de chiffrement pour AWS KMS

Pour administrer votre magasin de clés AWS CloudHSM, AWS KMS se connecte au compte de l'utilisateur de chiffrement `kmsuser` (CU) du cluster sélectionné. Avant de créer votre magasin de clés AWS CloudHSM, vous devez créer le CU `kmsuser`. Ensuite, lorsque vous créez votre magasin de clés AWS CloudHSM, vous fournissez le mot de passe du `kmsuser` à AWS KMS. Chaque fois que vous connectez le magasin de clés AWS CloudHSM à son cluster AWS CloudHSM associé, AWS KMS se connecte en tant que `kmsuser` et effectue une rotation du mot de passe `kmsuser`.

> **Important**
>
> Ne spécifiez pas l'option 2FA lorsque vous créez l'`kmsuser`utilisateur de chiffrement. Si vous le faites, AWS KMS ne peut pas se connecter et votre magasin de clés AWS CloudHSM ne peut pas être connecté à ce cluster AWS CloudHSM. Une fois que vous spécifiez 2FA, vous ne pouvez pas l'annuler. Vous devez à la place supprimer l'utilisateur de chiffrement et le recréer.

Pour créer l'`kmsuser`utilisateur de chiffrement, utilisez la procédure suivante.

1. Démarrez cloudhsm_mgmt_util en suivant la procédure décrite dans la rubrique Getting started with CloudHSM Management Utility (CMU) [Démarrer avec CloudHSM Management Utility (CMU)] du Guide de l'utilisateur AWS CloudHSM.
2. Utilisez la commande createUser dans cloudhsm_mgmt_util pour créer un utilisateur de chiffrement nommé `kmsuser`. Le mot de passe doit contenir entre 7 et 32 caractères alphanumériques. Il est sensible à la casse et ne peut contenir aucun des caractères spéciaux.

   Par exemple, la commande suivante crée un CU `kmsuser` avec le mot de passe `kmsPswd`.

   ```
   aws-cloudhsm> createUser CU kmsuser kmsPswd
   ```

Créer un magasin de clés AWS CloudHSM (console)

Lorsque vous créez un magasin de clés AWS CloudHSM dans AWS Management Console, vous pouvez ajouter et créer les conditions requises comme partie intégrante de votre flux de travail. Toutefois, le processus est plus rapide que vous les avez assemblées au préalable.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), AWS CloudHSM key stores (Magasins de clés).
4. Choisissez Créer un magasin de clés.
5. Entrez un nom convivial pour le magasin de clés personnalisé. Le nom doit être unique parmi tous les magasins de clés personnalisés de votre compte.

   > **Important**
   >
   > N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

6. Sélectionnez un cluster AWS CloudHSM pour le magasin de clés AWS CloudHSM. Sinon, pour créer un nouveau cluster AWS CloudHSM, choisissez le lien Create an AWS CloudHSM cluster (Créer un cluster HSM).

   Le menu affiche les clusters AWS CloudHSM de votre compte et de votre région qui ne sont pas déjà associés à un magasin de clés AWS CloudHSM. Le cluster doit respecter les exigences d'association à un magasin de clés personnalisé.
7. Choisissez Choose file (Choisir un fichier), puis chargez le certificat de point de confiance pour le cluster AWS CloudHSM que vous avez choisi. Il s'agit du fichier `customerCA.crt` que vous avez créé lorsque vous avez initialisé le cluster.
8. Entrez le mot de passe de l'utilisateur de chiffrement `kmsuser` (CU) que vous avez créé dans le cluster sélectionné.
9. Choisissez Créer.

Lorsque la procédure est réussie, le nouveau magasin de clés AWS CloudHSM s'affiche dans la liste des magasins de clés AWS CloudHSM du compte et de la région. S'il ne réussit pas, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez Dépannage d'un magasin de clés personnalisé.

Si vous essayez de créer un magasin de clés AWS CloudHSM avec toutes les mêmes valeurs de propriétés qu'un magasin de clés AWS CloudHSM déconnecté existant, AWS KMS ne crée pas de magasin de clés AWS CloudHSM et ne lève pas d'exception ou d'erreur. Au lieu de cela, AWS KMS reconnaît que le doublon est la conséquence probable d'une nouvelle tentative et renvoie l'ID du magasin de clés AWS CloudHSM existant.

Suivant : les nouveaux magasins de clés AWS CloudHSM ne sont pas automatiquement connectés. Avant de pouvoir créer les AWS KMS keys dans le magasin de clés AWS CloudHSM, vous devez connecter le magasin de clés personnalisé à son cluster AWS CloudHSM associé.

Créer un magasin de clés AWS CloudHSM (API)

Vous pouvez utiliser cette CreateCustomKeyStoreopération pour créer un nouveau magasin de AWS CloudHSM clés associé à un AWS CloudHSM cluster dans le compte et la région. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

L'opération `CreateCustomKeyStore` nécessite les valeurs de paramètre suivantes.

- CustomKeyStoreName — Un nom convivial pour le magasin de clés personnalisé, unique dans le compte.

  > **Important**
  >
  > N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

- CloudHsmClusterId — L'ID de cluster d'un AWS CloudHSM cluster répondant aux exigences d'un magasin de AWS CloudHSM clés.
- KeyStorePassword — Le mot de passe du compte `kmsuser` CU dans le cluster spécifié.
- TrustAnchorCertificate — Le contenu du `customerCA.crt` fichier que vous avez créé lors de l'initialisation du cluster.

L'exemple suivant utilise un ID de cluster fictif. Avant d'exécuter la commande, remplacez-le par un ID de cluster valide.

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleCloudHSMKeyStore \
        --cloud-hsm-cluster-id cluster-1a23b4cdefg \
        --key-store-password kmsPswd \
        --trust-anchor-certificate <certificate-goes-here>
```

Si vous utilisez l'AWS CLI, vous pouvez spécifier le fichier de certificat « trust anchor », au lieu de son contenu. Dans l'exemple suivant, le fichier `customerCA.crt` se trouve dans le répertoire racine.

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleCloudHSMKeyStore \
        --cloud-hsm-cluster-id cluster-1a23b4cdefg \
        --key-store-password kmsPswd \
        --trust-anchor-certificate file://customerCA.crt
```

Lorsque l'opération est réussie, `CreateCustomKeyStore` renvoie l'ID du magasin de clés personnalisé, comme illustré dans l'exemple de réponse suivant.

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Si l'opération échoue, corrigez l'erreur indiquée par l'exception, puis réessayez. Pour obtenir de l'aide supplémentaire, consultez Dépannage d'un magasin de clés personnalisé.

Si vous essayez de créer un magasin de clés AWS CloudHSM avec toutes les mêmes valeurs de propriétés qu'un magasin de clés AWS CloudHSM déconnecté existant, AWS KMS ne crée pas de magasin de clés AWS CloudHSM et ne lève pas d'exception ou d'erreur. Au lieu de cela, AWS KMS reconnaît que le doublon est la conséquence probable d'une nouvelle tentative et renvoie l'ID du magasin de clés AWS CloudHSM existant.

Suivant : pour utiliser le magasin de clés AWS CloudHSM, connectez-le à son cluster AWS CloudHSM.

Afficher un magasin de clés AWS CloudHSM

Vous pouvez consulter les AWS CloudHSM principaux magasins de chaque compte et région à l'aide de la AWS KMS console ou de l'DescribeCustomKeyStoresopération.

Voir aussi :

- Afficher un magasin de clés externe
- Afficher les clés KMS dans un magasin de clés AWS CloudHSM
- Journalisation des appels d'API AWS KMS avec AWS CloudTrail

Rubriques

- Afficher un magasin de clés AWS CloudHSM (console)
- Afficher un magasin de clés AWS CloudHSM (API)

Afficher un magasin de clés AWS CloudHSM (console)

Lorsque vous consultez les magasins de clés AWS CloudHSM dans la AWS Management Console, vous pouvez voir les éléments suivants :

- Nom et ID du magasin de clés personnalisé
- ID du cluster AWS CloudHSM associé.
- Nombre de modules HSM dans le cluster
- État actuel de la connexion

Un état de connexion (Status [État]) dont la valeur est Disconnected (Déconnecté) indique que le magasin de clés personnalisé est nouveau et n'a jamais été connecté, ou qu'il a été intentionnellement déconnecté de son cluster AWS CloudHSM. Toutefois, si vos tentatives d'utiliser une clé KMS dans un magasin de clés personnalisé connecté échouent, cela peut signifier la présence d'un problème avec le magasin de clés personnalisé ou son cluster AWS CloudHSM. Pour obtenir de l'aide, veuillez consulter Comment corriger les clés KMS défaillantes.

Pour afficher les magasins de clés AWS CloudHSM d'un compte et d'une région donnés, utilisez la procédure suivante.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), AWS CloudHSM key stores (Magasins de clés).

Pour personnaliser l'affichage, cliquez sur l'icône d'engrenage qui apparaît sous le bouton Créer un magasin de clés.

Afficher un magasin de clés AWS CloudHSM (API)

Pour consulter vos AWS CloudHSM principaux magasins, utilisez l'DescribeCustomKeyStoresopération. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Toutefois, vous pouvez utiliser le paramètre `CustomKeyStoreName` ou `CustomKeyStoreId` (mais pas les deux) pour limiter la sortie à un magasin de clés personnalisé en particulier. Pour les magasins de clés AWS CloudHSM, la sortie comprend l'ID et le nom du magasin de clés personnalisé, le type de magasin de clés personnalisé, l'ID du cluster AWS CloudHSM associé et l'état de la connexion. Si l'état de la connexion indique une erreur, la sortie inclut également un code d'erreur qui décrit la raison de l'erreur.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Par exemple, la commande suivante renvoie tous les magasins de clés personnalisées du compte et de la région. Vous pouvez utiliser les paramètres `Marker` et `Limit` pour parcourir les magasins de clés personnalisés de la sortie.

```
$ aws kms describe-custom-key-stores
```

L'exemple de commande suivant utilise le paramètre `CustomKeyStoreName` pour obtenir uniquement le magasin de clés personnalisé avec le nom convivial `ExampleCloudHSMKeyStore`. Vous pouvez utiliser le paramètre `CustomKeyStoreName` ou le paramètre `CustomKeyStoreId` (mais pas les deux) dans chaque commande.

L'exemple de sortie suivant représente un magasin de clés AWS CloudHSM qui est connecté à son cluster AWS CloudHSM.

> **Note**
>
> Le champ `CustomKeyStoreType` a été ajouté à la réponse `DescribeCustomKeyStores` pour distinguer les magasins de clés AWS CloudHSM des magasins de clés externes.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleCloudHSMKeyStore
{
```

```
   "CustomKeyStores": [
      {
         "CloudHsmClusterId": "cluster-1a23b4cdefg",
         "ConnectionState": "CONNECTED",
         "CreationDate": "1.499288695918E9",
         "CustomKeyStoreId": "cks-1234567890abcdef0",
         "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
         "CustomKeyStoreType": "AWS_CLOUDHSM",
         "TrustAnchorCertificate": "<certificate appears here>"
      }
   ]
}
```

Si `ConnectionState` a la valeur `Disconnected`, cela indique que le magasin de clés personnalisé n'a jamais été connecté ou qu'il a été intentionnellement déconnecté de son cluster AWS CloudHSM. Toutefois, si vos tentatives d'utiliser une clé KMS dans un magasin de clés AWS CloudHSM connecté échouent, cela peut signifier la présence d'un problème avec le magasin de clés AWS CloudHSM ou son cluster AWS CloudHSM. Pour obtenir de l'aide, veuillez consulter Comment corriger les clés KMS défaillantes.

Si `ConnectionState` a la valeur `FAILED`, la réponse `DescribeCustomKeyStores` inclut un élément `ConnectionErrorCode` qui explique la raison de l'erreur.

Par exemple, dans la sortie suivante, la valeur `INVALID_CREDENTIALS` indique que la connexion du magasin de clés personnalisé a échoué, car le mot de passe `kmsuser` n'est pas valide. Pour obtenir de l'aide sur ce sujet et sur d'autres échecs de connexion, consultez Dépannage d'un magasin de clés personnalisé.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      {
         "CloudHsmClusterId": "cluster-1a23b4cdefg",
         "ConnectionErrorCode": "INVALID_CREDENTIALS",
         "ConnectionState": "FAILED",
         "CustomKeyStoreId": "cks-1234567890abcdef0",
         "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
         "CustomKeyStoreType": "AWS_CLOUDHSM",
         "CreationDate": "1.499288695918E9",
         "TrustAnchorCertificate": "<certificate appears here>"
      }
   ]
```

```
}
```

Modifier les paramètres d'un magasin de clés AWS CloudHSM

Vous pouvez modifier les paramètres d'un magasin de clés AWS CloudHSM existant. Le magasin de clés personnalisé doit être déconnecté de son cluster AWS CloudHSM.

Pour modifier les paramètres du magasin de clés AWS CloudHSM :

1. Déconnectez le magasin de clés personnalisé de son cluster AWS CloudHSM. Même si le magasin de clés personnalisé est déconnecté, vous ne pouvez pas créer de AWS KMS keys (clés KMS) dans le magasin de clés personnalisé et vous ne pouvez pas utiliser les clés KMS qu'il contient pour les opérations de chiffrement.
2. Modifiez un ou plusieurs paramètres du magasin de clés AWS CloudHSM.
3. Reconnectez le magasin de clés personnalisé à son cluster AWS CloudHSM.

Vous pouvez modifier les paramètres suivants d'un magasin de clés personnalisé :

Le nom convivial du magasin de clés personnalisé.

Entrez un nouveau nom convivial. Le nouveau nom doit être unique parmi tous vos magasins de clés personnalisés de votre Compte AWS.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

L'ID de cluster du cluster AWS CloudHSM associé.

Modifiez cette valeur pour remplacer un cluster AWS CloudHSM associé par celui d'origine. Vous pouvez utiliser cette fonction pour réparer un magasin de clés personnalisé si son cluster AWS CloudHSM devient corrompu ou est supprimé.

Spécifiez un cluster AWS CloudHSM qui partage un historique des sauvegardes avec le cluster d'origine et répond aux exigences d'association à un magasin de clés personnalisé, y compris deux modules HSM actifs dans différentes zones de disponibilité. Les clusters qui partagent un historique des sauvegardes historique ont le même certificat de cluster. Pour afficher le certificat de cluster d'un cluster, utilisez l'DescribeClustersopération. Vous ne pouvez pas utiliser

la fonctionnalité de modification pour associer le magasin de clés personnalisé à un cluster AWS CloudHSM sans relation.

Mot de passe actuel de l'**`kmsuser`** utilisateur de chiffrement (CU).

Indique à AWS KMS le mot de passe actuel de l'utilisateur de chiffrement `kmsuser` dans le cluster AWS CloudHSM. Cette action ne permet pas de modifier le mot de passe de l'utilisateur de chiffrement `kmsuser` dans le cluster AWS CloudHSM.

Si vous modifiez le mot de passe de l'utilisateur de chiffrement (CU) `kmsuser` dans le cluster AWS CloudHSM, utilisez cette fonctionnalité pour informer AWS KMS du nouveau mot de passe de `kmsuser`. Dans le cas contraire, AWS KMS peut pas se connecter au cluster et toutes les tentatives pour connecter le magasin de clés personnalisé au cluster échouent.

Rubriques

- Modifier un magasin de clés AWS CloudHSM (console)
- Modifier un magasin de clés AWS CloudHSM (API)

Modifier un magasin de clés AWS CloudHSM (console)

Lorsque vous modifiez un magasin de clés AWS CloudHSM, vous pouvez modifier une ou plusieurs des valeurs configurables.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), AWS CloudHSM key stores (Magasins de clés).
4. Sélectionnez la ligne du magasin de clés AWS CloudHSM que vous souhaitez modifier.

   Si la valeur de la colonne Status n'est pas Disconnected, vous devez déconnecter le magasin de clés personnalisé avant de pouvoir le modifier. (Dans le menu Key store actions (Actions de magasin de clés), choisissez Disconnect [Déconnecter].)

   Même si un magasin de clés AWS CloudHSM est déconnecté, vous pouvez gérer le magasin de clés AWS CloudHSM et ses clés KMS, mais vous ne pouvez pas créer ou utiliser des clés KMS dans le magasin de clés AWS CloudHSM.

5. À partir du menu Key store actions (Actions de magasin de clés), choisissez Edit (Modifier).
6. Effectuez une ou plusieurs des actions suivantes :

   - Entrez un nouveau nom convivial pour le magasin de clés personnalisé.
   - Entrez l'ID de cluster d'un cluster AWS CloudHSM associé.
   - Saisissez le mot de passe de l'utilisateur du chiffrement `kmsuser` dans le cluster AWS CloudHSM associé.
7. Choisissez Enregistrer.

   Quand la procédure est réussie, un message décrit les paramètres que vous avez modifiés. Si elle ne réussit pas, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez Dépannage d'un magasin de clés personnalisé.
8. Reconnectez le magasin de clés personnalisé.

   Pour utiliser le magasin de clés AWS CloudHSM, vous devez le reconnecter après la modification. Vous pouvez laisser le magasin de clés AWS CloudHSM déconnecté. Mais tant qu'il est déconnecté, vous ne pouvez pas créer de clés KMS dans le magasin de clés AWS CloudHSM ni utiliser les clés KMS du magasin de clés AWS CloudHSM dans des opérations cryptographiques.

Modifier un magasin de clés AWS CloudHSM (API)

Pour modifier les propriétés d'un magasin de AWS CloudHSM clés, utilisez l'UpdateCustomKeyStoreopération. Vous pouvez modifier plusieurs propriétés d'un magasin de clés personnalisé dans la même commande. Si l'opération aboutit, AWS KMS renvoie une réponse HTTP 200 et un objet JSON sans propriétés. Pour vérifier que les modifications sont effectives, utilisez l'DescribeCustomKeyStoresopération.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Commencez par utiliser DisconnectCustomKeyStorepour déconnecter le magasin de clés personnalisé de son AWS CloudHSM cluster. Remplacez l'exemple d'ID de magasin de clés personnalisé, cks-1234567890abcdef0, par un ID réel.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Le premier exemple utilise UpdateCustomKeyStorepour remplacer le nom convivial du magasin de AWS CloudHSM clés par`DevelopmentKeys`. La commande utilise le paramètre `CustomKeyStoreId` pour identifier le magasin de clés AWS CloudHSM et le paramètre `CustomKeyStoreName` pour spécifier le nouveau nom du magasin de clés personnalisé.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-custom-key-store-name DevelopmentKeys
```

L'exemple suivant remplace le cluster associé à un magasin de clés AWS CloudHSM par une autre sauvegarde du même cluster. La commande utilise le paramètre `CustomKeyStoreId` pour identifier le magasin de clés AWS CloudHSM et le paramètre `CloudHsmClusterId` pour spécifier le nouvel ID de cluster.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

L'exemple suivant indique à AWS KMS que le mot de passe de `kmsuser` est `ExamplePassword`. La commande utilise le paramètre `CustomKeyStoreId` pour identifier le magasin de clés AWS CloudHSM et le paramètre `KeyStorePassword` pour spécifier le mot de passe actuel.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

La commande finale reconnecte le magasin de clés AWS CloudHSM à son cluster AWS CloudHSM. Vous pouvez laisser le magasin de clés personnalisé à l'état déconnecté, mais vous devez le connecter avant de pouvoir créer des clés KMS ou d'utiliser les clés KMS existantes pour les opérations de chiffrement. Remplacez l'exemple d'ID de magasin de clés personnalisé par un ID réel.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Connecter et déconnecter un magasin de clés AWS CloudHSM

Les nouveaux magasins de clés AWS CloudHSM ne sont pas connectés. Avant de pouvoir créer et utiliser les AWS KMS keys de votre magasin de clés AWS CloudHSM, vous devez le connecter à son cluster AWS CloudHSM associé. Vous pouvez connecter et déconnecter votre magasin de clés AWS CloudHSM à tout moment, et afficher son état de connexion.

Vous n'avez pas besoin de connecter votre magasin de clés AWS CloudHSM. Vous pouvez conserver un magasin de clés AWS CloudHSM dans un état déconnecté indéfiniment et le connecter

uniquement lorsque vous avez besoin de l'utiliser. Cependant, vous pouvez tester la connexion régulièrement pour vérifier que les paramètres sont corrects et que le magasin peut être connecté.

> **Note**
>
> Les magasins de clés AWS CloudHSM sont à l'état de connexion `DISCONNECTED` uniquement lorsque le magasin de clés n'a jamais été connecté ou que vous le déconnectez explicitement. Si votre magasin de clés AWS CloudHSM est à l'état de connexion `CONNECTED`, mais que vous rencontrez des difficultés à l'utiliser, assurez-vous que son cluster AWS CloudHSM associé est actif et contient au moins un module HSM actif. Pour obtenir de l'aide concernant les connexions ayant échoué, veuillez consulter the section called “Dépannage d'un magasin de clés personnalisé”.

Rubriques

- Connecter un magasin de clés AWS CloudHSM
- Déconnecter un magasin de clés AWS CloudHSM
- Connecter un magasin de clés AWS CloudHSM (console)
- Connecter un magasin de clés personnalisé (API)
- Déconnecter un magasin de clés AWS CloudHSM (console)
- Déconnecter un magasin de clés AWS CloudHSM (API)

Connecter un magasin de clés AWS CloudHSM

Lorsque vous connectez un magasin de clés AWS CloudHSM, AWS KMS recherche le cluster AWS CloudHSM associé, s'y connecte, se connecte au client AWS CloudHSM comme utilisateur de chiffrement `kmsuser` (CU), puis effectue une rotation du mot de passe `kmsuser`. AWS KMS reste connecté au client AWS CloudHSM aussi longtemps que le magasin de clés AWS CloudHSM est connecté.

Pour établir la connexion, AWS KMS crée un groupe de sécurité nommé `kms-<custom key store ID>` dans le cloud privé virtuel (VPC) du cluster. Le groupe de sécurité possède une règle qui permet le trafic entrant depuis le groupe de sécurité du cluster. AWS KMS crée également une interface réseau Elastic (ENI) dans chaque zone de disponibilité du sous-réseau privé pour le cluster. AWS KMS ajoute les interfaces réseau Elastic (ENI) au `kms-<cluster ID>` groupe de sécurité

et au groupe de sécurité du cluster. La description de chaque ENI est `KMS managed ENI for cluster <cluster-ID>`.

Le processus de connexion peut prendre un certain temps pour s'achever, jusqu'à 20 minutes.

Avant de connecter le magasin de clés AWS CloudHSM, vérifiez qu'il répond aux exigences.

- Son cluster AWS CloudHSM cluster associé doit contenir au moins un module HSM actif. Pour connaître le nombre de HSM dans le cluster, visualisez le cluster dans la AWS CloudHSM console ou utilisez l'DescribeClustersopération. Si nécessaire, vous pouvez ajouter un module HSM.
- Le cluster doit avoir un compte d'utilisateur de chiffrement `kmsuser` (CU), mais cet CU ne peut pas être connecté au cluster lorsque vous connectez le magasin de clés AWS CloudHSM. Pour obtenir de l'aide sur la déconnexion, reportez-vous à la section Comment se déconnecter et se reconnecter.
- L'état de connexion du magasin de clés AWS CloudHSM ne peut pas être `DISCONNECTING` ou `FAILED`. Pour afficher l'état de la connexion, utilisez la AWS KMS console ou la DescribeCustomKeyStoresréponse. Si l'état de la connexion est `FAILED`, déconnectez le magasin de clés personnalisé, résolvez le problème, puis connectez-le à nouveau.

Pour obtenir de l'aide concernant les connexions ayant échoué, veuillez consulter Comment corriger un échec de connexion.

Lorsque votre magasin de clés AWS CloudHSM est connecté, vous pouvez y créer des clés KMS et utiliser les clés KMS existantes dans les opérations cryptographiques.

Déconnecter un magasin de clés AWS CloudHSM

Lorsque vous déconnectez un magasin de clés AWS CloudHSM, AWS KMS se déconnecte du client AWS CloudHSM, se déconnecte du cluster AWS CloudHSM associé et supprime l'infrastructure réseau qu'il a créée pour prendre en charge la connexion.

Même si un magasin de clés AWS CloudHSM est déconnecté, vous pouvez gérer le magasin de clés AWS CloudHSM et ses clés KMS, mais vous ne pouvez pas créer ou utiliser des clés KMS dans le magasin de clés AWS CloudHSM. L'état de connexion du magasin de clés est `DISCONNECTED` et l'état de la clé des clés KMS du magasin de clés personnalisé est `Unavailable`, sauf si elles sont `PendingDeletion`. Vous pouvez reconnecter le magasin de clés AWS CloudHSM à tout moment.

Lorsque vous déconnectez un magasin de clés personnalisé, les clés KMS du magasin de clés deviennent immédiatement inutilisables (sous réserve d'une éventuelle cohérence). Toutefois, les

ressources chiffrées à l'aide de clés de données protégées par la clé KMS ne sont pas affectées tant que la clé KMS n'est pas réutilisée, par exemple pour déchiffrer la clé de données. Ce problème affecte les Services AWS, dont beaucoup utilisent des clés de données pour protéger vos ressources. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

> **Note**
>
> Même si un magasin de clés personnalisé est déconnecté, toutes les tentatives de création de clés KMS dans le magasin de clés personnalisé ou d'utilisation de clés KMS existantes dans les opérations de chiffrement échouent. Cette action peut empêcher les utilisateurs de stocker des données sensibles et d'y accéder.

Pour mieux estimer l'effet de la déconnexion de votre magasin de clés personnalisé, identifiez les clés KMS du magasin de clés personnalisé et déterminez leur utilisation antérieure.

Vous pouvez déconnecter un magasin de clés AWS CloudHSM pour des raisons telles que les suivantes :

- Pour effectuer une rotation du mot de passe **`kmsuser`**. AWS KMS modifie le mode de passe de `kmsuser` chaque fois qu'il se connecte au cluster AWS CloudHSM. Pour forcer une rotation de mot de passe, déconnectez-vous et reconnectez-vous.
- Pour auditer les éléments de clé des clés KMS du cluster AWS CloudHSM. Lorsque vous déconnectez le magasin de clés personnalisé, AWS KMS se déconnecte du compte de l'utilisateur de chiffrement `kmsuser` du client AWS CloudHSM. Ceci vous permet de vous connecter au cluster en tant qu'utilisateur du chiffrement `kmsuser`, et d'auditer et gérer les éléments de clé pour la clé KMS.
- Pour désactiver immédiatement toutes les clés KMS dans le magasin de clés AWS CloudHSM. Vous pouvez désactiver et réactiver les clés KMS dans un magasin de AWS CloudHSM clés en utilisant l'DisableKeyopération AWS Management Console ou. Ces opérations s'effectuent rapidement, mais elles agissent sur une seule clé KMS à la fois. La déconnexion du magasin de clés AWS CloudHSM fait immédiatement passer l'état de toutes les clés KMS du magasin de clés AWS CloudHSM à `Unavailable`, ce qui les empêche d'être utilisées dans toute opération cryptographique.
- Pour réparer un échec de tentative de connexion. Si une tentative de connexion d'un magasin de clés AWS CloudHSM échoue (l'état de connexion du magasin de clés personnalisé est `FAILED`),

vous devez déconnecter le magasin de clés AWS CloudHSM avant d'essayer de le connecter à nouveau.

Connecter un magasin de clés AWS CloudHSM (console)

Pour connecter un magasin de clés AWS CloudHSM dans l'AWS Management Console, commencez par sélectionner le magasin de clés AWS CloudHSM à partir de la page Custom key stores (Magasins de clés personnalisés). Le processus de connexion peut prendre jusqu'à 20 minutes.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), AWS CloudHSM key stores (Magasins de clés).
4. Choisissez la ligne du magasin de clés AWS CloudHSM que vous souhaitez connecter.

   Si l'état de connexion du magasin de clés AWS CloudHSM est Failed (Échec), vous devez déconnecter le magasin de clés personnalisé avant de le connecter.
5. Dans le menu Key store actions (Actions de magasin de clés), choisissez Connect (Connecter).

AWS KMS commence le processus de connexion de votre magasin de clés personnalisé. Il recherche le cluster AWS CloudHSM associé, crée l'infrastructure réseau requise, la connecte, se connecte au cluster AWS CloudHSM en tant qu'utilisateur de chiffrement (CU) `kmsuser` et effectue une rotation du mot de passe `kmsuser`. Une fois l'opération terminée, l'état de la connexion devient Connected.

Si l'opération échoue, un message d'erreur s'affiche qui décrit la raison de l'échec. Avant d'essayer de le connecter à nouveau, affichez l'état de connexion de votre magasin de clés AWS CloudHSM. Si le statut est Failed, vous devez déconnecter le magasin de clés personnalisé avant de vous connecter à nouveau. Si vous avez besoin d'aide, consultez Dépannage d'un magasin de clés personnalisé.

Suivant : the section called “Créer des clés KMS dans un magasin de clés AWS CloudHSM”.

Connecter un magasin de clés personnalisé (API)

Pour connecter un magasin de AWS CloudHSM clés déconnecté, utilisez l'ConnectCustomKeyStoreopération. Le cluster AWS CloudHSM associé doit contenir au moins un module HSM actif et l'état de la connexion ne peut pas être `FAILED`.

Le processus de connexion peut prendre un certain temps pour s'achever, jusqu'à 20 minutes. Sauf si elle échoue rapidement, l'opération renvoie une réponse HTTP 200 et un objet JSON sans propriétés. Cependant, cette réponse initiale n'indique pas que la connexion a abouti. Pour déterminer l'état de connexion du magasin de clés personnalisé, consultez la DescribeCustomKeyStoresréponse.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Pour identifier le magasin de clés AWS CloudHSM, utilisez son ID du magasin de clés personnalisé. Vous pouvez trouver l'ID sur la page des stockages de clés personnalisés de la console ou en utilisant l'DescribeCustomKeyStoresopération sans paramètres. Avant d'exécuter cet exemple, remplacez l'ID de l'exemple par un ID valide.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Pour vérifier que le magasin de AWS CloudHSM clés est connecté, utilisez l'DescribeCustomKeyStoresopération. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Toutefois, vous pouvez utiliser le paramètre `CustomKeyStoreName`ou `CustomKeyStoreId` (mais pas les deux) pour limiter la réponse à des magasins de clés personnalisés en particulier. Si `ConnectionState` a la valeur `CONNECTED`, cela indique que le magasin de clés personnalisé est connecté à son cluster AWS CloudHSM.

> **Note**
>
> Le champ `CustomKeyStoreType` a été ajouté à la réponse `DescribeCustomKeyStores` pour distinguer les magasins de clés AWS CloudHSM des magasins de clés externes.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
```

```
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

Si la valeur de `ConnectionState` est failed, l'élément `ConnectionErrorCode` indique la raison de l'échec. Dans ce cas, AWS KMS n'a pas pu trouver un cluster AWS CloudHSM dans votre compte avec l'ID de cluster `cluster-1a23b4cdefg`. Si vous avez supprimé le cluster, vous pouvez le restaurer à partir d'une sauvegarde du cluster d'origine, puis modifier l'ID de cluster pour le magasin de clés personnalisé. Pour obtenir de l'aide afin de répondre à un code d'erreur de connexion, veuillez consulter la rubrique Comment corriger un échec de connexion.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
      "ConnectionErrorCode": "CLUSTER_NOT_FOUND"
   ],
}
```

Suivant : Créer des clés KMS dans un magasin de clés AWS CloudHSM.

Déconnecter un magasin de clés AWS CloudHSM (console)

Pour déconnecter un magasin de clés AWS CloudHSM connecté dans la AWS Management Console, commencez à sélectionner le magasin de clés AWS CloudHSM à partir de la page Custom Key Stores (Magasins de clés personnalisés).

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.

2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), AWS CloudHSM key stores (Magasins de clés).
4. Choisissez la ligne du magasin de clés externe que vous souhaitez déconnecter.
5. Dans le menu Key store actions (Actions de magasin de clés), choisissez Disconnect (Déconnecter).

Une fois l'opération terminée, l'état de la connexion passe de Disconnecting à Disconnected. Si l'opération échoue, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez Dépannage d'un magasin de clés personnalisé.

Déconnecter un magasin de clés AWS CloudHSM (API)

Pour déconnecter un magasin de AWS CloudHSM clés connecté, utilisez l'DisconnectCustomKeyStoreopération. Si l'opération aboutit, AWS KMS renvoie une réponse HTTP 200 et un objet JSON sans propriétés.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Cet exemple déconnecte un magasin de clés AWS CloudHSM. Avant d'exécuter cet exemple, remplacez l'ID de l'exemple par un ID valide.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Pour vérifier que le magasin de AWS CloudHSM clés est déconnecté, utilisez l'DescribeCustomKeyStoresopération. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Toutefois, vous pouvez utiliser le paramètre `CustomKeyStoreName` ou `CustomKeyStoreId` (mais pas les deux) pour limiter la réponse à des magasins de clés personnalisés en particulier. Si `ConnectionState` a la valeur `DISCONNECTED`, cela indique que cet exemple de magasin de clés AWS CloudHSM n'est pas connecté à son cluster AWS CloudHSM.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
```

```
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "1.499288695918E9",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>"
   ],
}
```

Supprimer un magasin de clés AWS CloudHSM

Lorsque vous supprimez un magasin de clés AWS CloudHSM, AWS KMS supprime toutes les métadonnées relatives au magasin de clés AWS CloudHSM à partir de KMS, y compris les informations sur son association à un cluster AWS CloudHSM. Cette opération n'a pas d'incidence sur le cluster AWS CloudHSM, ses modules HSM ou ses utilisateurs. Vous pouvez créer un magasin de clés AWS CloudHSM qui est associé au même cluster AWS CloudHSM, mais vous ne pouvez pas annuler l'opération de suppression.

Vous ne pouvez supprimer un magasin de clés AWS CloudHSM que s'il est déconnecté de son cluster AWS CloudHSM et ne contient aucunes AWS KMS keys. Avant de supprimer un magasin de clés personnalisé, procédez comme suit :

- Vérifiez que vous n'aurez jamais besoin d'utiliser l'une des clés KMS du magasin de clés pour des opérations de chiffrement. Ensuite, planifiez la suppression de toutes les clés KMS du magasin de clés. Pour obtenir de l'aide sur la recherche des clés KMS dans un magasin de clés AWS CloudHSM, veuillez consulter la rubrique Rechercher les clés KMS d'un magasin de clés AWS CloudHSM.
- Vérifiez que toutes les clés KMS ont été supprimées. Pour afficher les clés KMS dans un magasin de clés AWS CloudHSM, veuillez consulter la rubrique Afficher les clés KMS dans un magasin de clés AWS CloudHSM.
- Déconnectez le magasin de clés AWS CloudHSM de son cluster AWS CloudHSM.

Au lieu de supprimer le magasin de clés AWS CloudHSM, pensez à le déconnecter de son cluster AWS CloudHSM associé. Même si un magasin de clés AWS CloudHSM est déconnecté, vous pouvez gérer le magasin de clés AWS CloudHSM et ses AWS KMS keys. Toutefois, vous ne pouvez pas créer ou utiliser de clés KMS dans le magasin de clés AWS CloudHSM. Vous pouvez reconnecter le magasin de clés AWS CloudHSM à tout moment.

Rubriques

- Supprimer un magasin de clés AWS CloudHSM (console)
- Supprimer un magasin de clés AWS CloudHSM (API)

Supprimer un magasin de clés AWS CloudHSM (console)

Pour supprimer un magasin de clés AWS CloudHSM dans l'AWS Management Console, commencez par sélectionner le magasin de clés AWS CloudHSM à partir de la page Custom key stores (Magasins de clés personnalisés).

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), AWS CloudHSM key stores (Magasins de clés).
4. Recherchez la ligne qui représente le magasin de clés AWS CloudHSM que vous souhaitez supprimer. Si Connection state (État de connexion) du magasin de clés AWS CloudHSM n'est pas Disconnected (Déconnecté), vous devez déconnecter le magasin de clés AWS CloudHSM avant de le supprimer.
5. Dans le menu Key store actions (Actions de magasin de clés), choisissez Delete (Supprimer).

Une fois l'opération terminée, un message de réussite s'affiche et le magasin de clés AWS CloudHSM n'apparaît plus dans la liste des magasins de clés. Si l'opération échoue, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez Dépannage d'un magasin de clés personnalisé.

Supprimer un magasin de clés AWS CloudHSM (API)

Pour supprimer un magasin de AWS CloudHSM clés, utilisez l'DeleteCustomKeyStoreopération. Si l'opération aboutit, AWS KMS renvoie une réponse HTTP 200 et un objet JSON sans propriétés.

Pour commencer, vérifiez que le magasin de clés AWS CloudHSM ne contient aucunes AWS KMS keys. Vous ne pouvez pas supprimer un magasin de clés personnalisé qui contient des clés KMS. Le premier exemple de commande utilise ListKeysset DescribeKeypour rechercher AWS KMS keys dans le magasin de clés avec l'exemple d'ID de magasin de AWS CloudHSM clés personnalisé

cks-1234567890abcdef0. Dans ce cas, la commande ne renvoie aucune clé KMS. Si c'est le cas, utilisez l'ScheduleKeyDeletionopération pour planifier la suppression de chacune des clés KMS.

Bash

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

Ensuite, déconnectez le magasin de clés AWS CloudHSM. Cet exemple de commande utilise l'DisconnectCustomKeyStoreopération pour déconnecter un magasin de AWS CloudHSM clés de son AWS CloudHSM cluster. Avant d'exécuter la commande, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Une fois le magasin de clés personnalisé déconnecté, vous pouvez utiliser DeleteCustomKeyStorecette opération pour le supprimer.

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Gérer les clés KMS dans un magasin de clés CloudHSM

Vous pouvez créer, afficher, gérer, utiliser et planifier la suppression des AWS KMS keys d'un magasin de clés AWS CloudHSM. Les procédures que vous employez sont très similaires à celles que vous utilisez pour les autres clés KMS. La seule différence est que vous spécifiez un magasin de clés AWS CloudHSM lorsque vous créez la clé KMS. Ensuite, AWS KMS crée un élément de clé non extractible pour la clé KMS dans le cluster AWS CloudHSM qui est associé au magasin de clés AWS CloudHSM. Lorsque vous utilisez une clé KMS dans un magasin de clés AWS CloudHSM, les opérations cryptographiques sont effectuées dans les modules HSM du cluster.

Fonctionnalités prises en charge

Outre les procédures décrites dans cette section, vous pouvez effectuer les actions suivantes avec les clés KMS d'un magasin de clés AWS CloudHSM :

- Utiliser les politiques de clé, les politiques IAM et les octrois pour autoriser l'accès aux clés KMS.
- Activer et désactiver les clés KMS.
- Attribuer des balises, créer des alias et utiliser le contrôle d'accès par attributs (ABAC) pour autoriser l'accès aux clés KMS.
- Utilisez les clés KMS pour les opérations de chiffrement, telles que le chiffrement, le déchiffrement, le rechiffrement et la génération de clés de données.
- Utiliser les clés KMS avec les services AWS qui s'intègrent à AWS KMS et prennent en charge les clés gérées par le client.
- Suivez l'utilisation de vos clés KMS dans les AWS CloudTrailjournaux et les outils CloudWatch de surveillance Amazon.

Fonctions non prises en charge

- Les magasins de clés AWS CloudHSM ne prennent en charge que les clés KMS de chiffrement symétriques. Vous ne pouvez pas créer de clés KMS HMAC, de clés KMS asymétriques ou de paires de clés de données asymétriques dans un magasin de clés AWS CloudHSM.
- Vous ne pouvez pas importer des éléments de clé dans une clé KMS dans un magasin de clés AWS CloudHSM. AWS KMS génère les éléments de clé pour la clé KMS dans le cluster AWS CloudHSM.
- Vous ne pouvez pas activer ou désactiver la rotation automatique des éléments de clé d'une clé KMS dans un magasin de clés AWS CloudHSM.

Rubriques

- Créer des clés KMS dans un magasin de clés AWS CloudHSM
- Afficher les clés KMS dans un magasin de clés AWS CloudHSM
- Utiliser les clés KMS dans un magasin de clés AWS CloudHSM
- Recherche des clés KMS et d'éléments de clé
- Planifier la suppression des clés KMS à partir d'un magasin de clés AWS CloudHSM

Créer des clés KMS dans un magasin de clés AWS CloudHSM

Une fois que vous avez créé un magasin de clés AWS CloudHSM, vous pouvez créer des AWS KMS keys dans votre magasin de clés. Il doit s'agir de clés KMS de chiffrement symétriques avec des éléments de clé générés par AWS KMS. Vous ne pouvez pas créer de clés KMS asymétriques, de clés KMS HMAC ou de clés KMS avec des éléments de clé importés dans un magasin de clé personnalisé. De plus, vous ne pouvez pas utiliser de clés KMS de chiffrement symétriques dans un magasin de clés personnalisé pour générer des paires de clés de données asymétriques.

Pour créer une clé KMS dans un magasin de clés AWS CloudHSM, le magasin de clés AWS CloudHSM doit être connecté au cluster AWS CloudHSM associé et le cluster doit contenir au moins deux modules HSM actifs dans différentes zones de disponibilité. Pour obtenir l'état de connexion et le nombre de modules HSM, affichez la page des magasins de clés AWS CloudHSM dans la AWS Management Console. Lorsque vous utilisez les opérations d'API, utilisez l'DescribeCustomKeyStoresopération pour vérifier que le magasin de AWS CloudHSM clés est connecté. Pour vérifier le nombre de HSM actifs dans le cluster et leurs zones de disponibilité, utilisez l'AWS CloudHSMDescribeClustersopération.

Lorsque vous créez une clé KMS dans votre magasin de clés AWS CloudHSM, AWS KMS crée la clé KMS dans AWS KMS. Toutefois, il crée les éléments de clé pour la clé KMS dans le cluster AWS CloudHSM associé. Plus précisément, AWS KMS se connecte au cluster comme le CU `kmsuser` que vous avez créé. Ensuite, il crée une clé symétrique AES 256 bits persistante, non extractible dans le cluster. AWS KMS définit la valeur de l'attribut de l'étiquette de clé, qui est visible uniquement dans le cluster, sur l'Amazon Resource Name (ARN) de la clé KMS.

Lorsque la commande réussit, l'état de la clé de la nouvelle clé KMS est `Enabled` et son origine est `AWS_CLOUDHSM`. Vous ne pouvez pas modifier l'origine d'une clé KMS après l'avoir créée. Lorsque vous affichez une clé KMS dans un magasin de AWS CloudHSM clés de la AWS KMS console ou en utilisant l'DescribeKeyopération, vous pouvez voir des propriétés typiques, telles que son identifiant

de clé, son état de clé et sa date de création. Mais vous pouvez également voir l'ID du magasin de clés personnalisé et l'ID du cluster AWS CloudHSM (facultatif). Pour plus de détails, consultez Afficher les clés KMS dans un magasin de clés AWS CloudHSM.

Si votre tentative de créer une clé KMS dans votre magasin de clés AWS CloudHSM échoue, référez-vous au message d'erreur pour vous aider à déterminer la cause. Il peut indiquer que le magasin de clés AWS CloudHSM n'est pas connecté (`CustomKeyStoreInvalidStateException`) ou que le cluster AWS CloudHSM associé ne possède pas les deux modules HSM actifs requis pour cette opération (`CloudHsmClusterInvalidConfigurationException`). Pour obtenir de l'aide, consultez Dépannage d'un magasin de clés personnalisé.

Pour un exemple de journal AWS CloudTrail de l'opération qui crée une clé KMS dans un magasin de clés AWS CloudHSM, veuillez consulter la rubrique CreateKey.

Rubriques

- Créer une clé KMS dans un magasin de clés AWS CloudHSM (console)
- Créer une clé KMS dans un magasin de clés AWS CloudHSM (API)

Créer une clé KMS dans un magasin de clés AWS CloudHSM (console)

Utilisez la procédure suivante pour créer une clé KMS de chiffrement symétrique dans un magasin de clés AWS CloudHSM.

> **Note**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key.

5. Choisissez Symmetric (Symétrique).
6. Dans Key usage (Utilisation de la clé), l'option Encrypt and decrypt (Chiffrer et déchiffrer) est sélectionnée pour vous. Ne la modifiez pas.
7. Choisissez Options avancées.
8. Dans le champ Key material origin (Origine de la clé), sélectionnez AWS CloudHSM key store (Magasin de clés).

   Vous ne pouvez pas créer de clés multi-régions dans un magasin de clés AWS CloudHSM.
9. Choisissez Suivant.
10. Sélectionnez un magasin de clés AWS CloudHSM pour votre nouvelle clé KMS. Pour créer un magasin de clés AWS CloudHSM, choisissez Create custom key store (Créer un magasin de clés personnalisé).

    Le magasin de clés AWS CloudHSM que vous sélectionnez doit avoir l'état Connected (Connecté). Son cluster AWS CloudHSM associé doit être actif et contenir au moins deux modules HSM dans différentes zones de disponibilité.

    Pour obtenir de l'aide concernant la connexion d'un magasin de clés AWS CloudHSM, veuillez consulter la rubrique Connecter et déconnecter un magasin de clés AWS CloudHSM. Pour obtenir de l'aide concernant l'ajout de modules HSM, veuillez consulter Ajout d'un module HSM dans le Guide de l'utilisateur AWS CloudHSM.
11. Choisissez Suivant.
12. Saisissez un alias et éventuellement une description pour la clé KMS.
13. (Facultatif). Sur la page Ajouter des identifications, ajoutez des identifications qui identifient ou catégorisent votre clé KMS.

    Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour de plus amples informations sur l'étiquetage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.
14. Choisissez Suivant.
15. Dans la section Administrateurs de clé, sélectionnez les utilisateurs et les rôles IAM qui peuvent gérer la clé KMS. Pour plus d'informations, veuillez consulter la rubrique Autorise les administrateurs de clé à administrer la clé KMS.

> **Note**
>
> Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation d'utiliser la clé KMS.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

16. (Facultatif) Pour empêcher les administrateurs de clé de supprimer cette clé KMS, décochez la case en bas de la page pour Autoriser les administrateurs de clé à supprimer cette clé.
17. Choisissez Suivant.
18. Dans la section Ce compte, sélectionnez les utilisateurs et rôles IAM de ce Compte AWS qui peuvent utiliser la clé KMS dans les opérations de chiffrement. Pour plus d'informations, veuillez consulter la rubrique Allows key users to use the KMS key (Autorise les utilisateurs de clé à utiliser la clé KMS).

> **Note**
>
> Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation d'utiliser la clé KMS.
> Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

19. (Facultatif) Vous pouvez autoriser d'autres Comptes AWS à utiliser cette clé KMS pour les opérations de chiffrement. Pour cela, dans la section Autres Comptes AWS en bas de la page, sélectionnez Ajouter un autre Compte AWS et saisissez l'ID Compte AWS d'un compte externe. Pour ajouter plusieurs comptes externes, répétez cette étape.

> **Note**
>
> Les administrateurs des autres Comptes AWS doivent également autoriser l'accès à la clé KMS en créant les politiques IAM pour leurs utilisateurs. Pour plus d'informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

20. Choisissez Suivant.
21. Passez en revue les paramètres de clé que vous avez choisis. Vous pouvez toujours revenir en arrière et modifier tous les paramètres.
22. Lorsque vous avez terminé, choisissez Finish (Terminer) pour créer la clé.

Lorsque la procédure réussit, l'affichage montre la nouvelle clé KMS dans le magasin de clés AWS CloudHSM que vous avez choisi. Lorsque vous choisissez le nom ou l'alias de la nouvelle clé KMS, l'onglet Cryptographic configuration (Configuration cryptographique) de sa page détaillée affiche l'origine de la clé KMS (AWS CloudHSM), le nom, l'ID et le type du magasin de clés personnalisé, ainsi que l'ID du cluster AWS CloudHSM. Si la procédure échoue, un message d'erreur s'affiche qui décrit l'échec.

> **Tip**
>
> Pour faciliter l'identification des clés KMS dans un magasin de clés personnalisé, sur la page Clés gérées par le client, ajoutez la colonne Custom key store ID (ID du magasin de clés personnalisé) à l'affichage. Cliquez sur l'icône des paramètres en haut à droite et sélectionnez ID du magasin de clés personnalisé. Pour plus de détails, consultez Personnalisation de vos tables de clés KMS.

Créer une clé KMS dans un magasin de clés AWS CloudHSM (API)

Pour créer une nouvelle AWS KMS key(clé KMS) dans votre magasin de AWS CloudHSM clés, utilisez l'CreateKeyopération. Utilisez le paramètre `CustomKeyStoreId` pour identifier votre magasin de clés personnalisé et spécifier une valeur `Origin` égale à `AWS_CLOUDHSM`.

Vous pouvez également souhaiter utiliser le paramètre `Policy` pour spécifier une politique de clé. Vous pouvez modifier la politique clé (PutKeyPolicy) et ajouter des éléments facultatifs, tels qu'une description et des balises à tout moment.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

L'exemple suivant commence par un appel à l'DescribeCustomKeyStoresopération visant à vérifier que le magasin de AWS CloudHSM clés est connecté au AWS CloudHSM cluster associé. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Pour décrire uniquement un magasin de clés AWS CloudHSM spécifique, utilisez son paramètre `CustomKeyStoreId` ou `CustomKeyStoreName` (mais pas les deux).

Avant d'exécuter cette commande, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

> **Note**
>
> N'incluez pas d'informations confidentielles ou sensibles dans les champs `Description` ou `Tags`. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS CloudHSM key store",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

L'exemple de commande suivant utilise l'DescribeClustersopération pour vérifier que le AWS CloudHSM cluster associé au `ExampleKeyStore` (cluster-1a23b4cdefg) possède au moins deux HSM actifs. Si le cluster a moins de deux HSM, l'opération `CreateKey` échoue.

```
$ aws cloudhsmv2 describe-clusters
{
    "Clusters": [
        {
```

```
            "SubnetMapping": {
               ...
            },
            "CreateTimestamp": 1507133412.351,
            "ClusterId": "cluster-1a23b4cdefg",
            "SecurityGroup": "sg-865af2fb",
            "HsmType": "hsm1.medium",
            "VpcId": "vpc-1a2b3c4d",
            "BackupPolicy": "DEFAULT",
            "Certificates": {
                "ClusterCertificate": "-----BEGIN CERTIFICATE-----\...\n-----END
 CERTIFICATE-----\n"
            },
            "Hsms": [
                {
                    "AvailabilityZone": "us-west-2a",
                    "EniIp": "10.0.1.11",
                    "ClusterId": "cluster-1a23b4cdefg",
                    "EniId": "eni-ea8647e1",
                    "StateMessage": "HSM created.",
                    "SubnetId": "subnet-a6b10bd1",
                    "HsmId": "hsm-abcdefghijk",
                    "State": "ACTIVE"
                },
                {
                    "AvailabilityZone": "us-west-2b",
                    "EniIp": "10.0.0.2",
                    "ClusterId": "cluster-1a23b4cdefg",
                    "EniId": "eni-ea8647e1",
                    "StateMessage": "HSM created.",
                    "SubnetId": "subnet-b6b10bd2",
                    "HsmId": "hsm-zyxwvutsrqp",
                    "State": "ACTIVE"
                },
            ],
            "State": "ACTIVE"
        }
    ]
}
```

Cet exemple de commande utilise l'CreateKeyopération pour créer une clé KMS dans un magasin de AWS CloudHSM clés. Pour créer une clé KMS dans un magasin de clés AWS CloudHSM, vous

devez fournir l'ID du magasin de clés personnalisé AWS CloudHSM et spécifier une valeur `Origin` pour `AWS_CLOUDHSM`.

La réponse inclut les ID du magasin de clés personnalisé et le cluster AWS CloudHSM.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

```
$ aws kms create-key --origin AWS_CLOUDHSM --custom-key-store-id cks-1234567890abcdef0
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1.499288695918E9,
    "Description": "Example key",
    "Enabled": true,
    "MultiRegion": false,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_CLOUDHSM"
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CustomKeyStoreId": "cks-1234567890abcdef0"
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Afficher les clés KMS dans un magasin de clés AWS CloudHSM

Pour afficher les AWS KMS keys dans un magasin de clés AWS CloudHSM, utilisez les mêmes techniques que vous utiliseriez pour afficher n'importe quelles clés AWS KMS gérées par le client. Consultez le Affichage des clés pour en savoir plus. Pour identifier les clés de votre cluster AWS CloudHSM qui font office de clés pour votre clé KMS, veuillez consulter Recherche des clés KMS et d'éléments de clé. Pour plus d'informations sur l'affichage des journaux AWS CloudTrail qui enregistrent toutes les opérations d'API dans un magasin de clés personnalisé, consultez Journalisation des appels d'API AWS KMS avec AWS CloudTrail.

Dans la console AWS KMS, les clés KMS de votre magasin de clés personnalisé sont affichées sur la page des clés gérées par le client en même temps que toutes les autres clés gérées par le client sur votre Compte AWS et dans votre région.

Toutefois, les valeurs suivantes sont spécifiques aux clés KMS d'un magasin de clés AWS CloudHSM.

- Le nom et l'ID du magasin de clés AWS CloudHSM qui stocke la clé KMS.
- L'ID de cluster du cluster AWS CloudHSM associé qui contient leurs clés.
- Une `Origin` dont la valeur est `AWS CloudHSM` dans la console AWS KMS ou `AWS_CLOUDHSM` dans les réponses d'API.
- La valeur de l'état de la clé peut être `Unavailable`. Pour obtenir de l'aide sur la résolution du statut, consultez Comment corriger les clés KMS non disponibles.

Afficher les clés KMS d'un magasin de clés AWS CloudHSM (Console)

1. Ouvrez la console AWS KMS à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Dans le coin supérieur droit, choisissez l'icône d'engrenage, choisissez ID de magasin de clés personnalisé et Origine, puis choisissez Confirmer.
5. Pour identifier les clés KMS d'un magasin de clés AWS CloudHSM, recherchez les clés KMS dont le champ Origin (Origine) a la valeur AWS CloudHSM. Pour identifier les clés KMS d'un magasin de clés AWS CloudHSM, affichez les valeurs de la colonne Custom key store ID (ID de magasin de clés personnalisé).
6. Choisissez l'alias ou l'ID de clé d'une clé KMS dans un magasin de clés AWS CloudHSM.

   Cette page affiche des informations détaillées sur la clé KMS, notamment son Amazon Resource Name (ARN), sa politique de clé et ses balises.
7. Choisissez l'onglet Cryptographic configuration (Configuration de chiffrement). Les onglets se trouvent sous la section General configuration (Configuration générale).

   Cette section contient des informations sur le magasin de clés AWS CloudHSM et le cluster AWS CloudHSM associé aux clés KMS.

Afficher les clés KMS d'un magasin de clés personnalisé (API)

Vous utilisez les mêmes opérations d'AWS KMSAPI pour afficher les clés KMS dans un magasin de AWS CloudHSM clés que vous utiliseriez pour n'importe quelle clé KMS ListKeys, y compris DescribeKey, et GetKeyPolicy. Par exemple, l'opération `describe-key` suivante de l'AWS CLI affiche les champs spéciaux d'une clé KMS dans un magasin de clés AWS CloudHSM. Avant d'exécuter une telle commande, remplacez l'exemple d'ID de clé KMS par une valeur valide.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

{
   "KeyMetadata": {
      "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "AWSAccountId": "111122223333",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CreationDate": 1537582718.431,
      "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "Description": "Key in custom key store",
      "Enabled": true,
      "EncryptionAlgorithms": [
         "SYMMETRIC_DEFAULT"
      ],
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyManager": "CUSTOMER",
      "KeySpec": "SYMMETRIC_DEFAULT",
      "KeyState": "Enabled",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "MultiRegion": false,
      "Origin": "AWS_CLOUDHSM"
   }
}
```

Pour obtenir de l'aide sur la recherche de clés KMS dans un magasin de clés AWS CloudHSM ou l'identification des clés de votre cluster AWS CloudHSM qui font office d'éléments de clé pour votre clé KMS, veuillez consulter la rubrique Recherche des clés KMS et d'éléments de clé.

Utiliser les clés KMS dans un magasin de clés AWS CloudHSM

Après avoir créé une clé KMS de chiffrement symétrique dans un magasin de clés AWS CloudHSM, vous pouvez l'utiliser pour les opérations cryptographiques suivantes :

- Encrypt
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Les opérations qui génèrent des paires de clés de données asymétriques GenerateDataKeyPairet GenerateDataKeyPairWithoutPlaintextne sont pas prises en charge dans les magasins de clés personnalisés.

Lorsque vous utilisez votre clé KMS dans une requête, vous devez identifier la clé KMS par son ID ou alias ; vous n'avez pas besoin de spécifier le magasin de clés AWS CloudHSM ou le cluster AWS CloudHSM. La réponse inclut les mêmes champs qui sont renvoyés pour une clé KMS de chiffrement symétrique.

Toutefois, lorsque vous utilisez une clé KMS dans un magasin de clés AWS CloudHSM, l'opération cryptographique est effectuée entièrement au sein du cluster AWS CloudHSM associé au magasin de clés AWS CloudHSM. L'opération utilise les éléments de clé du cluster associés à la clé KMS que vous avez choisie.

Pour que cela soit possible, les conditions suivantes sont requises.

- L'état de la clé KMS doit être `Enabled`. Pour trouver l'état clé, utilisez le champ Status de la AWS KMSconsole ou le `KeyState` champ de la DescribeKeyréponse.
- Le magasin de clés AWS CloudHSM doit être connecté à son cluster AWS CloudHSM. Son statut dans la AWS KMSconsole ou `ConnectionState` dans la DescribeCustomKeyStoresréponse doit être`CONNECTED`.
- Le cluster AWS CloudHSM associé au magasin de clés personnalisé doit contenir au moins un module HSM. Pour connaître le nombre de HSM actifs dans le cluster, utilisez la AWS KMSconsole, la AWS CloudHSM console ou l'DescribeClustersopération.
- Le cluster AWS CloudHSM doit contenir les éléments de clé de la clé KMS. Si la clé a été supprimé du cluster, ou qu'un module HSM a été créé à partir d'une sauvegarde qui n'inclut pas la clé de chiffrement, l'opération de chiffrement échoue.

Si ces conditions ne sont pas satisfaites, l'opération de chiffrement échoue, et AWS KMS renvoie une exception `KMSInvalidStateException`. En général, vous devez simplement reconnecter

le magasin de clés AWS CloudHSM. Pour obtenir de l'aide supplémentaire, consultez Comment corriger les clés KMS défaillantes.

Lorsque vous utilisez les clés KMS dans un magasin de clés AWS CloudHSM, sachez que les clés KMS de chaque magasin de clés AWS CloudHSM partagent un quota de requête de magasin de clés personnalisé pour les opérations cryptographiques. Si vous dépassez le quota, AWS KMS renvoie un `ThrottlingException`. Si le cluster AWS CloudHSM associé au magasin de clés AWS CloudHSM traite de nombreuses commandes, y compris celles non liées au magasin de clés AWS CloudHSM, une `ThrottlingException` peut être levée à un taux plus faible que prévu. Si vous obtenez une exception `ThrottlingException` pour une demande, réduisez la fréquence des demandes et essayez les commandes à nouveau. Pour plus d'informations sur le quota de magasin de clés personnalisé, veuillez consulter la rubrique Quotas de demandes de magasin de clés personnalisé.

Recherche des clés KMS et d'éléments de clé

Si vous gérez un magasin de clés AWS CloudHSM, il se peut que vous ayez besoin d'identifier les clés KMS dans chaque magasin de clés AWS CloudHSM. Par exemple, il se peut que vous ayez besoin de faire certaines tâches suivantes.

- Suivre les clés KMS du magasin de clés AWS CloudHSM dans les journaux AWS CloudTrail.
- Prédire l'effet de la déconnexion d'un magasin de clés AWS CloudHSM sur les clés KMS.
- Planifier la suppression des clés KMS avant de supprimer un magasin de clés AWS CloudHSM.

De plus, vous pouvez identifier les clés de votre cluster AWS CloudHSM qui font office de clés pour vos clés KMS. Bien que AWS KMS gère les clés KMS et leurs clés, vous devez toujours conserver le contrôle et la responsabilité de la gestion de votre cluster AWS CloudHSM, ainsi que de son HSM et des sauvegardes et des clés du module HSM. Il se peut que vous ayez besoin d'identifier les clés afin de contrôler les éléments de clé, de les protéger contre les suppressions accidentelles ou de les supprimer des HSM et des sauvegardes de clusters après avoir supprimé la clé KMS.

Tous les éléments de clé des clés KMS de votre magasin de clés AWS CloudHSM sont détenus par l'utilisateur de chiffrement `kmsuser` (CU). AWS KMS définit l'attribut d'étiquette clé, qui n'est visible que dans AWS CloudHSM, sur l'Amazon Resource Name (ARN) de la clé KMS.

Pour rechercher les clés KMS et les éléments de clé, utilisez l'une des techniques suivantes.

- Rechercher les clés KMS d'un magasin de clés AWS CloudHSM : comment identifier les clés KMS dans un ou tous vos magasins de clés AWS CloudHSM.

- Rechercher toutes les clés d'un magasin de clés AWS CloudHSM : comment trouver toutes les clés de votre cluster qui font office d'éléments de clé pour les clés KMS dans votre magasin de clés AWS CloudHSM.
- Rechercher la clé AWS CloudHSM pour une clé KMS : comment trouver la clé de votre cluster qui fait office d'élément de clé pour une clé KMS particulière de votre magasin de clés AWS CloudHSM.
- Rechercher la clé KMS pour une clé AWS CloudHSM — Comment trouver la clé KMS pour une clé particulière de votre cluster.

Rechercher les clés KMS d'un magasin de clés AWS CloudHSM

Si vous gérez un magasin de clés AWS CloudHSM, il se peut que vous ayez besoin d'identifier les clés KMS dans chaque magasin de clés AWS CloudHSM. Ces informations vous permettent de suivre les opérations de la clé KMS dans les journaux AWS CloudTrail, de prédire l'effet de la déconnexion d'un magasin de clés personnalisé sur les clés KMS ou de planifier la suppression des clés KMS avant de supprimer un magasin de clés AWS CloudHSM.

Rechercher les clés KMS d'un magasin de clés AWS CloudHSM (console)

Pour rechercher les clés KMS d'un magasin de clés AWS CloudHSM spécifique, sur la page Customer managed keys (Clés gérées par le client), affichez les valeurs des champs Custom Key Store Name (Nom du magasin de clés personnalisé) ou Custom Key Store ID (ID de magasin de clés personnalisé). Pour identifier les clés KMS d'un magasin de clés AWS CloudHSM, recherchez les clés KMS dont le champ Origin (Origine) a la valeur AWS CloudHSM. Pour ajouter des colonnes facultatives à l'affichage, choisissez l'icône d'engrenage dans le coin supérieur droit de la page.

Rechercher les clés KMS d'un magasin de clés AWS CloudHSM (API)

Pour rechercher les clés KMS dans un magasin de AWS CloudHSM clés, utilisez les DescribeKeyopérations ListKeyset filtrez par `CustomKeyStoreId` valeur. Avant d'exécuter les exemples, remplacez l'exemple d'ID de magasin de clés personnalisé fictif par une valeur valide.

Bash

Pour rechercher les clés KMS d'un magasin de clés AWS CloudHSM spécifique, obtenez l'ensemble de vos clés KMS de vos compte et région. Ensuite, filtrez sur l'ID de magasin de clés personnalisé.

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
```

```
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

Pour obtenir les clés KMS d'un magasin de clés AWS CloudHSM dans le compte et la région, recherchez les valeurs de `CustomKeyStoreType` équivalentes à `AWS_CloudHSM`.

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreType": "AWS_CloudHSM"' --context 100; done
```

PowerShell

Pour rechercher des clés KMS dans un magasin de AWS CloudHSM clés en particulier, utilisez les KmsKey applets de commande KmsKeyList Get - pour obtenir toutes vos clés KMS dans le compte et la région. Ensuite, filtrez sur l'ID de magasin de clés personnalisé.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

Pour obtenir des clés KMS dans n'importe quel magasin de AWS CloudHSM clés du compte et de la région, filtrez en fonction de la CustomKeyStoreType valeur de`AWS_CLOUDHSM`.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

Rechercher toutes les clés d'un magasin de clés AWS CloudHSM

Vous pouvez identifier les clés de votre cluster AWS CloudHSM qui font office d'éléments de clé pour votre magasin de clés AWS CloudHSM. Pour ce faire, utilisez la findAllKeyscommande dans cloudhs_mgmt_util pour rechercher les descripteurs de touches de toutes les clés détenues ou partagées. `kmsuser` À moins que vous ne vous soyez connecté en tant que `kmsuser` et que vous n'ayez créé des clés en dehors de AWS KMS, toutes les clés que `kmsuser` possède représentent des éléments de clé pour les clés KMS.

Tout responsable de chiffrement du cluster peut exécuter cette commande sans déconnecter le magasin de clés AWS CloudHSM.

1. Démarrez cloudhsm_mgmt_util en suivant la procédure décrite dans la rubrique Getting started with CloudHSM Management Utility (CMU) (Démarrer avec CloudHSM Management Utility [CMU]).

2. Connectez-vous à la commande cloudhsm_mgmt_util à l'aide d'un compte de responsable de chiffrement (CO).
3. Utilisez la commande listUsers pour trouver l'ID d'utilisateur de l' utilisateur de chiffrement `kmsuser`.

   Dans cet exemple, `kmsuser` a l'ID utilisateur 3.

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:3

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt         2FA
         1              PCO             admin                   NO
 0               NO
         2              AU              app_user                NO
 0               NO
         3              CU              kmsuser                 NO
 0               NO
```

4. Utilisez la findAllKeyscommande pour trouver les descripteurs de toutes les clés `kmsuser` détenues ou partagées. Remplacez l'exemple d'ID d'utilisateur (3) par l'ID réel d'utilisateur de `kmsuser` dans votre cluster.

   L'exemple de sortie montre que `kmsuser` possède des clés avec les descripteurs de clés 8, 9 et 262162 sur les deux HSM du cluster.

```
aws-cloudhsm> findAllKeys 3 0
Keys on server 0(10.0.0.1):
Number of keys found 3
number of keys matched from start index 0::6
8,9,262162
findAllKeys success on server 0(10.0.0.1)

Keys on server 1(10.0.0.2):
Number of keys found 6
number of keys matched from start index 0::6
8,9,262162
findAllKeys success on server 1(10.0.0.2)
```

Rechercher la clé KMS pour une clé AWS CloudHSM

Si vous connaissez le descripteur de clé d'une clé que `kmsuser` détient dans le cluster, vous pouvez utiliser l'étiquette de la clé pour identifier la clé KMS associée de votre magasin de clés AWS CloudHSM.

Lorsque AWS KMS crée les éléments de clé d'une clé KMS de votre cluster AWS CloudHSM, il écrit l'Amazon Resource Name (ARN) de la clé KMS dans l'étiquette de la clé. Sauf si vous avez modifié la valeur de l'étiquette, vous pouvez utiliser la commande getAttribute de key_mgmt_util ou cloudhsm_mgmt_util pour associer la clé à sa clé KMS.

Pour exécuter cette procédure, vous devez déconnecter temporairement le magasin de clés AWS CloudHSM afin de pouvoir vous connecter comme CU `kmsuser`.

> **Note**
>
> Même si un magasin de clés personnalisé est déconnecté, toutes les tentatives de création de clés KMS dans le magasin de clés personnalisé ou d'utilisation de clés KMS existantes dans les opérations de chiffrement échouent. Cette action peut empêcher les utilisateurs de stocker des données sensibles et d'y accéder.

1. Déconnectez le magasin de clés AWS CloudHSM, s'il n'est pas déjà déconnecté, puis connectez-vous à key_mgmt_util en tant que `kmsuser`, comme expliqué dans Comment se déconnecter et se connecter.
2. Utilisez la commande `getAttribute` dans key_mgmt_util ou cloudhsm_mgmt_util pour obtenir l'attribut d'étiquette (`OBJ_ATTR_LABEL`, attribut `3`) d'un descripteur de clé particulier.

   Par exemple, cette commande utilise `getAttribute` dans cloudhsm_mgmt_util pour obtenir l'attribut d'étiquette (attribut `3`) de la clé avec le descripteur de clé `262162`. La sortie montre que la clé `262162` sert d'éléments de clé pour la clé KMS avec l'ARN `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`. Avant d'exécuter cette commande, remplacez l'exemple de descripteur de clé par un descripteur valide.

   Pour obtenir la liste des attributs de clé, utilisez la commande listAttributes ou veuillez consulter la Référence des attributs de clé dans le Guide de l'utilisateur AWS CloudHSM.

   ```
   aws-cloudhsm> getAttribute 262162 3
   ```

```
Attribute Value on server 0(10.0.1.10):
OBJ_ATTR_LABEL
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

3. Déconnectez-vous de key_mgmt_util ou cloudhsm_mgmt_util et reconnectez le magasin de clés AWS CloudHSM comme expliqué dans Comment se déconnecter et se reconnecter.

Rechercher la clé AWS CloudHSM pour une clé KMS

Vous pouvez utiliser l'ID d'une clé KMS dans un magasin de clés AWS CloudHSM pour identifier la clé de votre cluster AWS CloudHSM qui fait office d'éléments de clé. Vous pouvez ensuite utiliser son descripteur de clé pour identifier la clé dans les commandes du client AWS CloudHSM.

Lorsque AWS KMS crée les éléments de clé d'une clé KMS de votre cluster AWS CloudHSM, il écrit l'Amazon Resource Name (ARN) de la clé KMS dans l'étiquette de la clé. Sauf si vous avez modifié la valeur, vous pouvez utiliser la commande findKey dans l'outil key_mgmt_util pour obtenir le descripteur de clé des éléments de clé de la clé KMS. Pour exécuter cette procédure, vous devez déconnecter temporairement le magasin de clés AWS CloudHSM afin de pouvoir vous connecter comme CU `kmsuser`.

> **Note**
>
> Même si un magasin de clés personnalisé est déconnecté, toutes les tentatives de création de clés KMS dans le magasin de clés personnalisé ou d'utilisation de clés KMS existantes dans les opérations de chiffrement échouent. Cette action peut empêcher les utilisateurs de stocker des données sensibles et d'y accéder.

1. Déconnectez le magasin de clés AWS CloudHSM, s'il n'est pas déjà déconnecté, puis connectez-vous à key_mgmt_util en tant que `kmsuser`, comme expliqué dans Comment se déconnecter et se connecter.
2. Utilisez la commande findKey de l'outil key_mgmt_util pour rechercher une clé avec une étiquette qui correspond à l'ARN d'une clé KMS de votre magasin de clés AWS CloudHSM. Remplacez l'exemple d'ARN de clé KMS dans la valeur du paramètre `-l` (L en lettres minuscules comme « label ») par un ARN de clé KMS valide.

   Par exemple, cette commande recherche la clé avec une étiquette qui correspond à l'exemple d'ARN d'une clé KMS, par exemple `arn:aws:kms:us-`

west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab. L'exemple de sortie montre que la clé avec le descripteur de clé 262162 possède l'ARN de la clé KMS spécifié dans son étiquette. Vous pouvez désormais utiliser ce descripteur de clé d'autres commandes key_mgmt_util.

```
Command: findKey -l arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
Total number of keys present 1

 number of keys matched from start index 0::1
262162

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

3. Déconnectez-vous de key_mgmt_util et reconnectez le magasin de clés personnalisé comme expliqué dans Comment se déconnecter et se reconnecter.

Planifier la suppression des clés KMS à partir d'un magasin de clés AWS CloudHSM

Lorsque vous avez la certitude que vous n'aurez pas besoin d'utiliser une AWS KMS key pour les opérations de chiffrement, vous pouvez planifier la suppression de la clé KMS. Utilisez la même procédure que vous utilisez pour planifier la suppression d'une clé KMS à partir de AWS KMS. En outre, gardez votre magasin de clés AWS CloudHSM connecté afin qu'AWS KMS puisse supprimer les éléments de clé correspondants du cluster AWS CloudHSM associé lorsque la période d'attente arrive à expiration.

Vous pouvez contrôler la planification, l'annulation et la suppression de la clé KMS dans vos journaux AWS CloudTrail.

> ⚠ Warning
>
> La suppression d'une clé KMS est une opération destructrice et potentiellement dangereuse, qui vous empêche de récupérer toutes les données chiffrées à l'aide de la clé KMS. Avant de planifier la suppression de la clé KMS, examinez l'utilisation passée de la clé KMS et créez une CloudWatch alarme Amazon qui vous avertit lorsque quelqu'un essaie d'utiliser la clé

KMS alors qu'elle est en attente de suppression. Chaque fois que possible, désactivez la clé KMS, au lieu de la supprimer.

Lorsque vous planifiez la suppression d'une clé KMS d'un magasin de clés AWS CloudHSM, son état passe à Pending deletion (Suppression en attente). La clé KMS reste à l'état Pending deletion (En attente de suppression) tout au long de la période d'attente, même si la clé KMS n'est pas disponible parce que vous avez déconnecté la clé personnalisée. Cela vous permet d'annuler la suppression de la clé KMS à tout moment au cours de la période d'attente.

Lorsque la période d'attente expire, AWS KMS supprime la clé KMS depuis AWS KMS. Ensuite, AWS KMS met tout en œuvre pour supprimer les éléments de clé du cluster AWS CloudHSM associé. Si AWS KMS ne peut pas supprimer les clés, comme lorsque, par exemple, le magasin de clés est déconnecté de AWS KMS, il se peut que vous ayez besoin de supprimer manuellement les clé orphelines du cluster.

AWS KMS ne supprime pas la clé des sauvegardes de clusters. Même si vous supprimez la clé KMS de AWS KMS et supprimez ses éléments de clé de votre cluster AWS CloudHSM, les clusters créés à partir des sauvegardes peuvent contenir la clé supprimée. Pour supprimer définitivement les éléments de clé, affichez la date de création de la clé KMS. Ensuite, supprimez toutes les sauvegardes de cluster qui peuvent contenir la clé.

Lorsque vous planifiez la suppression d'une clé KMS d'un magasin de clés AWS CloudHSM, la clé KMS devient immédiatement inutilisable (sous réserve d'une éventuelle cohérence). Toutefois, les ressources chiffrées à l'aide de clés de données protégées par la clé KMS ne sont pas affectées tant que la clé KMS n'est pas réutilisée, par exemple pour déchiffrer la clé de données. Ce problème affecte les Services AWS, dont beaucoup utilisent des clés de données pour protéger vos ressources. Pour plus d'informations, consultez Comment les clés KMS inutilisables affectent les clés de données.

Dépannage d'un magasin de clés personnalisé

Les magasins de clés AWS CloudHSM sont conçus pour être disponibles et résilients. Cependant, il existe certaines conditions d'erreur que vous pouvez avoir à réparer pour maintenir votre magasin de clés AWS CloudHSM opérationnel.

Rubriques

- Comment corriger les clés KMS non disponibles
- Comment corriger les clés KMS défaillantes

- Comment corriger un échec de connexion
- Comment répondre à un échec d'opération de chiffrement
- Comment corriger les informations d'identification kmsuser non valides
- Comment supprimer les éléments de clé orphelins
- Comment récupérer les éléments de clé supprimés pour une clé KMS
- Comment se connecter en tant que kmsuser

Comment corriger les clés KMS non disponibles

L'état de clé des AWS KMS keys dans un magasin de clés AWS CloudHSM est généralement `Enabled`. Comme toutes les clés KMS, l'état de clé change lorsque vous désactivez les clés KMS dans un magasin de clés AWS CloudHSM ou que vous programmez leur suppression. Toutefois, contrairement à d'autres clés KMS, les clés KMS d'un magasin de clés personnalisé peuvent également avoir un état de clé de `Unavailable`.

Un état de clé `Unavailable` indique que la clé KMS est dans un magasin de clés personnalisé qui a été intentionnellement déconnecté et que les tentatives pour le reconnecter, le cas échéant, ont échoué. Lorsqu'une clé KMS n'est pas disponible, vous pouvez afficher et gérer la clé KMS, mais vous ne pouvez pas l'utiliser dans les opérations de chiffrement.

Pour obtenir l'état de clé d'une clé KMS, sur la page Clés gérées par le client, veuillez consulter le champ Status (État) de la clé KMS. Vous pouvez également utiliser l'DescribeKeyopération et afficher l'`KeyState`élément dans la réponse. Pour plus de détails, veuillez consulter Affichage des clés.

Les clés KMS d'un magasin de clés personnalisé déconnecté possèdent l'état de clé `Unavailable` ou `PendingDeletion`. Les clés KMS dont la suppression a été planifiée à partir d'un magasin de clés personnalisé ont un état de clé `Pending Deletion`, même si le magasin de clés personnalisé est déconnecté. Cela vous permet d'annuler la suppression de clé planifiée sans reconnecter le magasin de clés personnalisé.

Pour corriger une clé KMS indisponible, reconnectez le magasin de clés personnalisé. Une fois le magasin de clés personnalisé reconnecté, l'état de clé des clés KMS du magasin de clés personnalisé est automatiquement restauré à son état précédent, comme `Enabled` ou `Disabled`. Les clés KMS qui sont en attente de suppression restent dans l'état `PendingDeletion`. Toutefois, si le problème persiste, l'activation et la désactivation d'une clé KMS indisponible ne changent pas son état. L'action d'activation ou de désactivation prend effet uniquement lorsque la clé devient disponible.

Pour obtenir de l'aide concernant les connexions ayant échoué, consultez Comment corriger un échec de connexion.

Comment corriger les clés KMS défaillantes

Les problèmes liés à la création et à l'utilisation des clés KMS dans les magasins de clés AWS CloudHSM peuvent être causés par un problème avec votre magasin de clés AWS CloudHSM, son cluster AWS CloudHSM associé, la clé KMS ou ses éléments de clé.

Lorsqu'un magasin de clés AWS CloudHSM est déconnecté de son cluster AWS CloudHSM, l'état de clé des clés KMS du magasin de clés personnalisé est `Unavailable`. Toutes les requêtes pour créer des clés KMS dans un magasin de clés AWS CloudHSM déconnecté renvoient une exception `CustomKeyStoreInvalidStateException`. Toutes les demandes pour chiffrer, déchiffrer, rechiffrer ou générer les clés de données renvoient une exception `KMSInvalidStateException`. Pour corriger le problème, reconnectez le magasin de clés AWS CloudHSM.

Toutefois, vos tentatives d'utiliser une clé KMS de magasin de clés AWS CloudHSM pour les opérations cryptographiques peuvent échouer même si son état de clé est `Enabled` et que l'état de connexion du magasin de clés AWS CloudHSM est `Connected`. Cela peut être dû à l'une des conditions suivantes.

- Les éléments de clé de la clé KMS peuvent avoir été supprimés du cluster AWS CloudHSM associé. Pour examiner, recherchez le handle de la clé des éléments de clé pour une clé KMS et, si nécessaire, essayez de récupérer les éléments de clés.
- Tous les modules HSM ont été supprimés du cluster AWS CloudHSM associé au magasin de clés AWS CloudHSM. Pour utiliser une clé KMS d'un magasin de clés AWS CloudHSM dans une opération cryptographique, son cluster AWS CloudHSM doit contenir au moins un module HSM actif. Pour vérifier le nombre et l'état des HSM dans un AWS CloudHSM cluster, utilisez la AWS CloudHSM console ou l'DescribeClustersopération. Pour ajouter un HSM au cluster, utilisez la AWS CloudHSM console ou l'CreateHsmopération.
- Le cluster AWS CloudHSM associé au magasin de clés AWS CloudHSM a été supprimé. Pour corriger le problème, créez un cluster à partir d'une sauvegarde qui est liée au cluster d'origine, telle qu'une sauvegarde du cluster d'origine ou une sauvegarde qui a été utilisée pour créer le cluster d'origine. Ensuite, modifiez l'ID de cluster dans les paramètres du magasin de clés personnalisé. Pour obtenir des instructions, veuillez consulter Comment récupérer les éléments de clé supprimés pour une clé KMS.
- Le cluster AWS CloudHSM associé au magasin de clés personnalisé ne disposait d'aucune session PKCS #11 disponible. Cela se produit généralement pendant les périodes de fort trafic

en rafale, lorsque des sessions supplémentaires sont nécessaires pour traiter le trafic. Pour réagir à une exception `KMSInternalException` avec un message d'erreur concernant les sessions PKCS #11, revenez en arrière et relancez la requête.

Comment corriger un échec de connexion

Si vous essayez de connecter un magasin de clés AWS CloudHSM à son cluster AWS CloudHSM, mais que l'opération échoue, l'état de connexion du magasin de clés AWS CloudHSM passe à `FAILED`. Pour connaître l'état de connexion d'un magasin de AWS CloudHSM clés, utilisez la AWS KMS console ou l'DescribeCustomKeyStoresopération.

Sinon, certains tentatives de connexion échouent rapidement en raison d'erreurs de configuration de cluster facilement détectées. Dans ce cas, l'état de la connexion est toujours `DISCONNECTED`. Ces échecs renvoient un message d'erreur ou une exception qui explique pourquoi la tentative a échoué. Vérifiez la description de l'exception et les exigences du cluster, corrigez le problème, mettez à jour le magasin de clés AWS CloudHSM si nécessaire, et essayez de le connecter à nouveau.

Lorsque l'état de connexion est `FAILED` défini, exécutez l'DescribeCustomKeyStoresopération et voyez l'`ConnectionErrorCode`élément dans la réponse.

> **Note**
>
> Si l'état de connexion du magasin de clés AWS CloudHSM est `FAILED`, vous devez déconnecter le magasin de clés AWS CloudHSM avant de le reconnecter. Vous ne pouvez pas connecter un magasin de clés AWS CloudHSM dont l'état de connexion est `FAILED`.

- `CLUSTER_NOT_FOUND` indique que AWS KMS ne peut pas trouver un cluster AWS CloudHSM avec l'ID de cluster spécifié. Cela peut se produire parce que le mauvais ID de cluster a été fourni à une opération d'API ou que le cluster a été supprimé et n'a pas été remplacé. Pour corriger cette erreur, vérifiez l'ID du cluster, par exemple à l'aide de la AWS CloudHSM console ou de l'DescribeClustersopération. Si le cluster a été supprimé, la créez un cluster à partir d'une sauvegarde récente de l'original. Ensuite, déconnectez le magasin de clés AWS CloudHSM, modifiez le paramètre d'ID de cluster du magasin de clés AWS CloudHSM et reconnectez le magasin de clés AWS CloudHSM au cluster.
- `INSUFFICIENT_CLOUDHSM_HSMS` indique que le cluster AWS CloudHSM associé ne contient aucun HSM. Pour vous connecter, le cluster doit avoir au moins un HSM. Pour trouver le nombre de HSM dans le cluster, utilisez l'DescribeClustersopération. Pour résoudre cette erreur, ajoutez

au moins un module HSM au cluster. Si vous ajoutez plusieurs HSM, il est préférable de les créer dans des zones de disponibilité différentes.

- `INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET` indique que AWS KMS n'a pas pu connecter le magasin de clés AWS CloudHSM à son cluster AWS CloudHSM, car au moins un sous-réseau privé associé au cluster n'a pas d'adresse IP disponible. Une connexion de magasin de clés AWS CloudHSM nécessite une adresse IP libre dans chacun des sous-réseaux privés associés, bien que deux soient préférables.

  Vous ne pouvez pas ajouter des adresses IP (blocs CIDR) vers un sous-réseau existant. Si possible, déplacez ou supprimez les autres ressources qui utilisent les adresses IP du sous-réseau, telles que les instances EC2 inutilisées ou les interfaces réseau Elastic. Sinon, vous pouvez créer un cluster à partir d'une sauvegarde récente du cluster AWS CloudHSM avec des sous-réseaux privés nouveaux ou existants qui ont plus d'espace d'adressage libre. Ensuite, pour associer le nouveau cluster à votre magasin de clés AWS CloudHSM, déconnectez le magasin de clés personnalisé, modifiez l'ID de cluster du magasin de clés AWS CloudHSM en lui affectant l'ID du nouveau cluster et essayez de le connecter à nouveau.

  > **Tip**
  >
  > Pour éviter de réinitialiser le mot de passe `kmsuser`, utilisez la sauvegarde la plus récente du cluster AWS CloudHSM.

- `INTERNAL_ERROR` indique que AWS KMS n'a pas pu terminer la demande en raison d'une erreur interne. Réitérez la requête . Pour les requêtes `ConnectCustomKeyStore`, déconnectez le magasin de clés AWS CloudHSM avant de tenter de le connecter à nouveau.
- `INVALID_CREDENTIALS` indique que AWS KMS ne peut pas se connecter au cluster AWS CloudHSM associé, car il ne dispose pas du mot de passe correct du compte `kmsuser`. Pour obtenir de l'aide sur cette erreur, veuillez consulter Comment corriger les informations d'identification `kmsuser` non valides.
- `NETWORK_ERRORS` indique généralement des problèmes réseau temporaires. Déconnectez le magasin de clés AWS CloudHSM, attendez quelques minutes et essayez de le connecter à nouveau.
- `SUBNET_NOT_FOUND` indique qu'au moins un sous-réseau de la configuration du cluster AWS CloudHSM a été supprimé. Si AWS KMS ne détecte pas l'ensemble des sous-réseaux dans la configuration du cluster, les tentatives de connexion du magasin de clés AWS CloudHSM au cluster AWS CloudHSM échouent.

Pour corriger cette erreur, créez un cluster à partir d'une sauvegarde récente du même cluster AWS CloudHSM. (Ce processus crée une nouvelle configuration de cluster avec un VPC et des sous-réseaux privés.) Vérifiez que le nouveau cluster répond aux conditions requises pour un magasin de clés personnalisé et notez l'ID du nouveau cluster. Ensuite, pour associer le nouveau cluster à votre magasin de clés AWS CloudHSM, déconnectez le magasin de clés personnalisé, modifiez l'ID de cluster du magasin de clés AWS CloudHSM en lui affectant l'ID du nouveau cluster et essayez de le connecter à nouveau.

> **Tip**
>
> Pour éviter de réinitialiser le mot de passe `kmsuser`, utilisez la sauvegarde la plus récente du cluster AWS CloudHSM.

- `USER_LOCKED_OUT` indique que le compte CU (utilisateur de chiffrement) `kmsuser` est verrouillé pour le cluster AWS CloudHSM associé en raison d'un trop grand nombre de tentatives de mot de passe ayant échoué. Pour obtenir de l'aide sur cette erreur, veuillez consulter Comment corriger les informations d'identification `kmsuser` non valides.

  Pour corriger cette erreur, déconnectez le magasin de clés AWS CloudHSM et utilisez la commande changePswd dans cloudhsm_mgmt_util pour modifier le mot de passe du compte `kmsuser`. Ensuite, modifiez le `kmsuser` paramètre de mot de passe pour le magasin de clés personnalisé, et essayez de vous connecter à nouveau. Pour obtenir de l'aide, utilisez la procédure décrite dans la rubrique Comment corriger les informations d'identification `kmsuser` non valides.
- `USER_LOGGED_IN` indique que le compte CU `kmsuser` est connecté au cluster AWS CloudHSM associé. Cela empêche AWS KMS de soumettre le mot de passe du compte `kmsuser` à une rotation et de se connecter au cluster. Pour corriger cette erreur, déconnectez le compte CU `kmsuser` du cluster. Si vous avez modifié le mot de passe `kmsuser` pour vous connecter au cluster, vous devez également mettre à jour la valeur du mot de passe du magasin de clés pour le magasin de clés AWS CloudHSM. Pour obtenir de l'aide, veuillez consulter Comment se déconnecter et se reconnecter.
- `USER_NOT_FOUND` indique que AWS KMS ne peut pas trouver de compte CU `kmsuser` dans le cluster AWS CloudHSM associé. Pour corriger cette erreur, créez un compte CU `kmsuser` dans le cluster, puis mettez à jour la valeur du mot de passe du magasin de clés AWS CloudHSM. Pour obtenir de l'aide, veuillez consulter Comment corriger les informations d'identification `kmsuser` non valides.

Comment répondre à un échec d'opération de chiffrement

Une opération de chiffrement qui utilise une clé KMS dans un magasin de clés personnalisé peut échouer avec un `KMSInvalidStateException`. Les messages d'erreur suivants peuvent accompagner le `KMSInvalidStateException`.

KMS ne peut pas communiquer avec votre cluster CloudHSM. Il peut s'agir d'un problème réseau transitoire. Si cette erreur s'affiche à plusieurs reprises, vérifiez que les ACL réseau et les règles de groupe de sécurité pour le VPC de votre cluster AWS CloudHSM sont correctes.

- Bien qu'il s'agisse d'une erreur HTTPS 400, elle peut résulter de problèmes réseau transitoires. Pour répondre, commencez par relancer la demande. Toutefois, si elle continue d'échouer, examinez la configuration de vos composants réseau. Cette erreur est probablement causée par une mauvaise configuration d'un composant réseau, telle qu'une règle de pare-feu ou une règle de groupe de sécurité VPC qui bloque le trafic sortant.

KMS ne peut pas communiquer avec votre cluster AWS CloudHSM car l'utilisateur kmsuser est verrouillé. Si cette erreur s'affiche à plusieurs reprises, déconnectez le magasin de clés AWS CloudHSM et réinitialisez le mot de passe du compte kmsuser. Mettez à jour le mot de passe kmsuser pour le magasin de clés personnalisé et réessayez la demande.

- Ce message d'erreur indique que le compte CU (utilisateur de chiffrement) `kmsuser` est verrouillé pour le cluster AWS CloudHSM associé en raison d'un trop grand nombre de tentatives de mot de passe ayant échoué. Pour obtenir de l'aide sur cette erreur, veuillez consulter Comment se déconnecter et se connecter.

Comment corriger les informations d'identification **`kmsuser`** non valides

Lorsque vous connectez un magasin de clés AWS CloudHSM, AWS KMS se connecte au cluster AWS CloudHSM associé en tant qu'utilisateur de chiffrement `kmsuser` (CU). Il reste identifié jusqu'à ce que le magasin de clés AWS CloudHSM soit déconnecté. La réponse DescribeCustomKeyStores réponse affiche pour `ConnectionState` la valeur `FAILED` et pour `ConnectionErrorCode` la valeur `INVALID_CREDENTIALS`, comme indiqué dans l'exemple suivant.

Si vous déconnectez le magasin de clés AWS CloudHSM et modifiez le mot de passe `kmsuser`, AWS KMS ne peut pas se connecter au cluster AWS CloudHSM avec les informations d'identification du compte CU `kmsuser`. Par conséquent, toutes les tentatives pour connecter le magasin de clés AWS CloudHSM échouent. La réponse `DescribeCustomKeyStores` réponse affiche pour `ConnectionState` la valeur `FAILED` et pour `ConnectionErrorCode` la valeur `INVALID_CREDENTIALS`, comme indiqué dans l'exemple suivant.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
   ],
}
```

De plus, au bout de cinq tentatives de connexion au cluster ayant échoué avec un mot de passe incorrect, AWS CloudHSM verrouille le compte utilisateur. Pour se connecter au cluster, vous devez modifier le mot de passe du compte.

Si AWS KMS obtient une réponse de verrouillage lorsqu'il tente de se connecter au cluster en tant que CU `kmsuser`, la requête de connexion du magasin de clés AWS CloudHSM échoue. La DescribeCustomKeyStoresréponse inclut un `ConnectionState` de `FAILED` et une `ConnectionErrorCode` valeur de`USER_LOCKED_OUT`, comme indiqué dans l'exemple suivant.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "USER_LOCKED_OUT"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
   ],
}
```

Pour réparer l'une ou l'autre de ces conditions, utilisez la procédure suivante.

1. Déconnectez le magasin de clés AWS CloudHSM.
2. Exécutez l'DescribeCustomKeyStoresopération et visualisez la valeur de l'`ConnectionErrorCode`eélément dans la réponse.

   - Si la valeur de `ConnectionErrorCode` est `INVALID_CREDENTIALS`, déterminez le mot de passe actuel pour le compte `kmsuser`. Si nécessaire, utilisez la commande changePswd dans cloudhsm_mgmt_util afin de définir le mot de passe pour une valeur connue.
   - Si la valeur de `ConnectionErrorCode` est `USER_LOCKED_OUT`, vous devez utiliser la commande changePswd dans cloudhsm_mgmt_util pour modifier le mot de passe `kmsuser`.
3. Modifiez le mot de passe actuel de `kmsuser` afin qu'il corresponde au mot de passe actuel de `kmsuser` dans le cluster. Cette action indique à AWS KMS le mot de passe à utiliser pour se connecter au cluster. Elle ne change pas le mot de passe de `kmsuser` dans le cluster.
4. Connectez le magasin de clés personnalisé.

Comment supprimer les éléments de clé orphelins

Après avoir planifié la suppression d'une clé KMS d'un magasin de clés AWS CloudHSM, vous devrez peut-être supprimer manuellement les éléments de clé correspondants du cluster AWS CloudHSM associé.

Lorsque vous créez une clé KMS dans un magasin de clés AWS CloudHSM, AWS KMS crée les métadonnées de la clé KMS dans AWS KMS et génère les éléments de clé dans le cluster AWS CloudHSM associé. Lorsque vous planifiez la suppression d'une clé KMS dans un magasin de clés AWS CloudHSM, AWS KMS supprime les métadonnées de la clé KMS à la fin de la période d'attente. Ensuite, AWS KMS met tout en œuvre pour supprimer les éléments de clé correspondants du cluster AWS CloudHSM. La tentative peut échouer si AWS KMS ne peut pas accéder au cluster, par exemple, lorsqu'il est déconnecté du magasin de clés AWS CloudHSM ou que le mot de passe `kmsuser` change. AWS KMS n'essaye pas de supprimer les éléments de clé des sauvegardes de cluster.

AWS KMS consigne les résultats de sa tentative de suppression des éléments de clé du cluster dans l'entrée d'événement `DeleteKey` de vos journaux AWS CloudTrail. Ils apparaissent dans l'élément `backingKeysDeletionStatus` de l'élément `additionalEventData`, comme illustré dans l'exemple d'entrée suivant. L'entrée inclut également l'ARN de clé KMS, l'ID de cluster AWS CloudHSM et le descripteur de clé des éléments de clé (`backing-key-id`).

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-12-10T14:23:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"keyHandle\":\"16\",\"backingKeyId\":
\"backing-key-id\",\"deletionStatus\":\"FAILURE\"}]"
    },
    "eventID": "c21f1f47-f52b-4ffe-bff0-6d994403cf40",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```


Pour supprimer les éléments de clé du cluster AWS CloudHSM associé, utilisez une procédure comme celle qui suit. Cet exemple utilise les outils de ligne de commande AWS CLI et AWS CloudHSM, mais vous pouvez utiliser AWS Management Console au lieu de la CLI.

1. Déconnectez le magasin de clés AWS CloudHSM, s'il n'est pas déjà déconnecté, puis connectez-vous à key_mgmt_util, comme expliqué dans Comment se déconnecter et se connecter.
2. Utilisez la commande deleteKey dans key_mgmt_util pour supprimer la clé à partir des modules HSM du cluster.

   Par exemple, cette commande supprime la clé 262162 à partir des modules HSM du cluster. Le descripteur de la clé est répertorié dans l'entrée du CloudTrail journal.

   ```
   Command: deleteKey -k 262162

           Cfm3DeleteKey returned: 0x00 : HSM Return: SUCCESS

           Cluster Error Status
           Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
           Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
           Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
   ```
3. Déconnectez-vous de key_mgmt_util et reconnectez le magasin de clés AWS CloudHSM comme expliqué dans Comment se déconnecter et se reconnecter.

Comment récupérer les éléments de clé supprimés pour une clé KMS

Si les éléments d'une AWS KMS key sont supprimés, la clé KMS est inutilisable et tout le texte chiffré qui l'a été sous la clé KMS ne peut pas être déchiffré. Cela peut se produire si les éléments de clé d'une clé KMS d'un magasin de clés AWS CloudHSM sont supprimés du cluster AWS CloudHSM associé. Toutefois, il peut être possible de récupérer les clés.

Lorsque vous créez une AWS KMS key (clé KMS) dans un magasin de clés AWS CloudHSM, AWS KMS se connecte au cluster AWS CloudHSM associé et crée les éléments de clé de la clé KMS. Il change également le mot de passe en une valeur que lui seul connaît et reste connecté tant que le magasin de clés AWS CloudHSM est connecté. Etant donné que seul le propriétaire des clés, à savoir, le CU qui a créé une clé, peut supprimer la clé, il est peu probable que la clé soit supprimé des modules HSM accidentellement.

Toutefois, si les éléments de clé d'une clé KMS sont supprimés des HSM dans un cluster, l'état de la clé KMS devient alors `UNAVAILABLE`. Si vous essayez d'utiliser la clé KMS pour une opération cryptographique, l'opération échoue avec une exception `KMSInvalidStateException`. Et surtout, toutes les données chiffrées à l'aide de la clé KMS ne peuvent pas être déchiffrées.

Dans certains cas, vous pouvez récupérer les clés supprimés par en créant un cluster à partir d'une sauvegarde qui contient les clés. Cette stratégie fonctionne uniquement lorsqu'au moins une sauvegarde a été créée, tandis que la clé existait et avant qu'elle n'ait été supprimée.

Utilisez la procédure suivante pour récupérer les éléments de clé.

1. Recherchez une sauvegarde de cluster qui contient les éléments de clé. La sauvegarde doit également contenir tous les utilisateurs et toutes les clés dont vous avez besoin pour prendre en charge le cluster et ses données chiffrées.

   Utilisez l'DescribeBackupsopération pour répertorier les sauvegardes d'un cluster. Utilisez ensuite l'horodatage de la sauvegarde afin de vous aider à sélectionner une sauvegarde. Pour limiter la sortie au cluster associé au magasin de clés AWS CloudHSM, utilisez le paramètre `Filters`, comme illustré dans l'exemple suivant.

   ```
   $ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
   {
       "Backups": [
           {
               "ClusterId": "cluster-1a23b4cdefg",
               "BackupId": "backup-9g87f6edcba",
               "CreateTimestamp": 1536667238.328,
               "BackupState": "READY"
           },
                ...
       ]
   }
   ```

2. Créez un cluster à partir de la sauvegarde sélectionnée. Vérifiez que la sauvegarde contient la clé supprimée et les clés que le cluster nécessite.
3. Déconnectez le magasin de clés AWS CloudHSM afin que vous puissiez modifier ses propriétés.
4. Modifiez l'ID de cluster du magasin de clés AWS CloudHSM. Entrez l'ID du cluster que vous avez créé à partir de la sauvegarde. Étant donné que le cluster partage un historique des sauvegardes avec le cluster d'origine, le nouvel ID de cluster doit être valide.
5. Reconnectez le magasin de clés AWS CloudHSM.

Comment se connecter en tant que **kmsuser**

Pour créer et gérer les éléments de clé dans le cluster AWS CloudHSM de votre magasin de clés AWS CloudHSM, AWS KMS utilise le compte de l'utilisateur de chiffrement kmsuser (CU). Vous créez le compte CU kmsuser dans votre cluster et fournissez son mot de passe à AWS KMS lorsque vous créez votre magasin de clés AWS CloudHSM.

En général, AWS KMS gère le compte kmsuser. Toutefois, pour certaines tâches, vous devez vous déconnecter du magasin de clés AWS CloudHSM, vous connecter au cluster en tant que CU kmsuser et utiliser les outils de ligne de commande cloudhsm_mgmt_util et key_mgmt_util.

> **Note**
>
> Même si un magasin de clés personnalisé est déconnecté, toutes les tentatives de création de clés KMS dans le magasin de clés personnalisé ou d'utilisation de clés KMS existantes dans les opérations de chiffrement échouent. Cette action peut empêcher les utilisateurs de stocker des données sensibles et d'y accéder.

Cette rubrique explique comment déconnecter votre magasin de clés AWS CloudHSM et vous connecter en tant que kmsuser, exécuter l'outil de ligne de commande AWS CloudHSM et déconnecter et reconnecter votre magasin de clés AWS CloudHSM.

Rubriques

- Comment se déconnecter et se connecter
- Comment se déconnecter et se reconnecter

Comment se déconnecter et se connecter

Utilisez la procédure suivante pour chaque fois que nécessaire pour vous connecter à un cluster associé en tant que CU kmsuser.

1. Déconnectez le magasin de clés AWS CloudHSM, s'il n'est pas déjà déconnecté. Vous pouvez utiliser la console AWS KMS ou l'API AWS KMS.

   Tant que votre clé AWS CloudHSM est connectée, AWS KMS est connecté en tant que kmsuser. Cela vous empêche de vous connecter comme kmsuser ou de modifier le mot de passe kmsuser.

Par exemple, cette commande permet DisconnectCustomKeyStorede déconnecter un exemple de magasin de clés. Remplacez l'exemple d'ID de magasin de clés AWS CloudHSM par un ID valide.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

2. Démarrez l'outil cloudhsm_mgmt_util. Utilisez la procédure décrite dans la section Préparation pour exécuter la commande cloudhsm_mgmt_util du Guide de l'utilisateur AWS CloudHSM.
3. Connectez-vous à cloudhsm_mgmt_util sur le cluster AWS CloudHSM comme responsable de chiffrement (CO).

   Par exemple, la commande suivante se connecte en tant qu'CO nommé admin. Remplacez l'exemple de nom d'utilisateur CO et le mot de passe par des valeurs valides.

```
aws-cloudhsm>loginHSM CO admin <password>
loginHSM success on server 0(10.0.2.9)
loginHSM success on server 1(10.0.3.11)
loginHSM success on server 2(10.0.1.12)
```

4. Utilisez la commande changePswd pour modifier le mot de passe du compte `kmsuser` en une valeur que vous connaissez. (AWS KMS effectue une rotation du mot de passe lorsque vous connectez votre magasin de clés AWS CloudHSM.) Le mot de passe doit contenir entre 7 et 32 caractères alphanumériques. Il est sensible à la casse et ne peut contenir aucun des caractères spéciaux.

   Par exemple, cette commande modifie le mot de passe de `kmsuser` en `tempPassword`.

```
aws-cloudhsm>changePswd CU kmsuser tempPassword

************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. Cav server does NOT synchronize these changes with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************

Do you want to continue(y/n)?y
Changing password for kmsuser(CU) on 3 nodes
```

5. Connectez-vous à l'outil key_mgmt_util ou cloudhsm_mgmt_util comme `kmsuser` à l'aide du mot de passe que vous avez défini. Pour obtenir des instructions détaillées, consultez la section Mise en route avec cloudhsm_mgmt_util et Mise en route avec l'outil key_mgmt_util. L'outil que vous choisissez dépend de votre tâche.

   Par exemple, cette commande se connecte à key_mgmt_util.

```
Command: loginHSM -u CU -s kmsuser -p tempPassword
Cfm3LoginHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Comment se déconnecter et se reconnecter

1. Exécutez la tâche, puis déconnectez-vous de l'outil de ligne de commande. Si vous ne vous déconnectez pas, les tentatives de reconnecter votre magasin de clés AWS CloudHSM échoueront.

```
Command:  logoutHSM
Cfm3LogoutHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

2. Modifiez le paramètre de mot de passe de `kmsuser` pour le magasin de clés personnalisé.

   Cela indique à AWS KMS le mot de passe actuel pour `kmsuser` dans le cluster. Si vous omettez cette étape, ne AWS KMS sera pas en mesure de se connecter au cluster comme `kmsuser` et toutes les tentatives de reconnecter votre magasin de clés personnalisé échoueront. Vous pouvez utiliser la AWS KMS console ou le `KeyStorePassword` paramètre de l'UpdateCustomKeyStoreopération.

   Par exemple, cette commande indique à AWS KMS que le mot de passe actuel est `tempPassword`. Remplacez l'exemple de mot de passe par le mot de passe réel.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --
key-store-password tempPassword
```

3. Reconnectez le magasin de clés AWS KMS à son cluster AWS CloudHSM. Remplacez l'exemple d'ID de magasin de clés AWS CloudHSM par un ID valide. Pendant le processus de connexion, AWS KMS modifie le mot de passe de `kmsuser` par une valeur que lui seul connaît.

   L'ConnectCustomKeyStoreopération revient rapidement, mais le processus de connexion peut prendre un certain temps. Cependant, cette réponse initiale n'indique pas que la connexion a réussi.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

4. Utilisez cette DescribeCustomKeyStoresopération pour vérifier que le magasin de AWS CloudHSM clés est connecté. Remplacez l'exemple d'ID de magasin de clés AWS CloudHSM par un ID valide.

   Dans cet exemple, le champ de l'état de connexion montre que le magasin de clés AWS CloudHSM est désormais connecté.

```
$ aws kms describe-custom-key-stores --custom-key-store-
id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

Magasins de clés externes

Les magasins de clés externes vous permettent de protéger vos ressources AWS en utilisant des clés cryptographiques en dehors d'AWS. Cette fonctionnalité avancée est conçue pour les charges de travail réglementées que vous devez protéger avec des clés de chiffrement stockées dans un système de gestion des clés externe que vous contrôlez. Les magasins de clés externes respectent

l'AWS Digital Sovereignty Pledge pour vous donner le contrôle souverain de vos données dans AWS, y compris la possibilité de chiffrer avec des éléments de clé que vous possédez et contrôlez en dehors d'AWS.

Un magasin de clés externe est un magasin de clés personnalisé soutenu par un gestionnaire de clés externe que vous possédez et contrôlez en dehors d'AWS. Votre gestionnaire de clés externe peut être un module de sécurité matérielle (HSM) physique ou virtuel, ou tout système matériel ou logiciel capable de générer et d'utiliser des clés cryptographiques. Les opérations de chiffrement et de déchiffrement qui utilisent une clé KMS dans un magasin de clés externe sont effectuées par votre gestionnaire de clés externe à l'aide de vos éléments de clé cryptographiques, une fonctionnalité connue sous le nom de Hold your Own Keys (HYOK).

AWS KMS n'interagit jamais directement avec votre gestionnaire de clés externe et ne peut pas créer, afficher, gérer ou supprimer vos clés. Au lieu de cela, AWS KMS n'interagit qu'avec le logiciel de proxy de magasin de clés externe (proxy XKS) que vous fournissez. Le proxy de votre magasin de clés externe assure la médiation de toutes les communications entre AWS KMS et votre gestionnaire de clés externe. Il transmet toutes les requêtes AWS KMS à votre gestionnaire de clés externe et transmet les réponses de votre gestionnaire de clés externe à AWS KMS. Le proxy de magasin de clés externe traduit également les requêtes génériques émises par AWS KMS dans un format propre au fournisseur que votre gestionnaire de clés externe peut comprendre, ce qui vous permet d'utiliser des magasins de clés externes avec des gestionnaires de clés provenant de différents fournisseurs.

Vous pouvez utiliser des clés KMS dans un magasin de clés externe pour le chiffrement côté client, notamment avec l'AWS Encryption SDK. Mais les magasins de clés externes constituent une ressource importante pour le chiffrement côté serveur, car elles vous permettent de protéger vos ressources AWS réparties sur différents Services AWS à l'aide de vos clés de chiffrement en dehors d'AWS. Les Services AWS qui prennent en charge les clés gérées par le client pour le chiffrement symétrique prennent également en charge les clés KMS d'un magasin de clés externes. Pour plus d'informations sur la prise en charge des services, consultez Intégration des services AWS.

Les magasins de clés externes vous permettent d'utiliser AWS KMS pour les charges de travail réglementées où les clés de chiffrement doivent être stockées et utilisées en dehors d'AWS. Ils constituent toutefois une rupture majeure par rapport au modèle standard de responsabilité partagée et nécessitent des charges opérationnelles supplémentaires. Pour la plupart des clients, le risque accru pour la disponibilité et la latence dépassera les avantages en termes de sécurité perçus pour les magasins de clés externes.

Les magasins de clés externes vous permettent de contrôler la source de confiance. Les données chiffrées au moyen des clés KMS dans votre magasin de clés externe ne peuvent être déchiffrées

qu'en utilisant le gestionnaire de clés externe que vous contrôlez. Si vous révoquez temporairement l'accès à votre gestionnaire de clés externe, par exemple en déconnectant le magasin de clés externe ou en déconnectant votre gestionnaire de clés externe du proxy de magasin de clés externe, AWS perd tout accès à vos clés cryptographiques jusqu'à ce que vous le rétablissiez. Pendant cet intervalle, le texte chiffré qui a été chiffré au moyen de vos clés KMS ne peut pas être déchiffré. Si vous révoquez définitivement l'accès à votre gestionnaire de clés externe, tout le texte chiffré au moyen d'une clé KMS dans votre magasin de clés externe devient irrécupérable. Les seules exceptions sont les services AWS qui mettent brièvement en cache les clés de données protégées par vos clés KMS. Ces clés de données continuent de fonctionner jusqu'à ce que vous désactiviez la ressource ou que le cache expire. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

Les magasins de clés externes débloquent les quelques cas d'utilisation liés aux charges de travail réglementées où les clés de chiffrement doivent rester sous votre seul contrôle et inaccessibles à AWS. Mais il s'agit d'un changement majeur dans la façon dont vous exploitez une infrastructure basée sur le cloud et d'un changement significatif du modèle de responsabilité partagée. Pour la plupart des charges de travail, la charge opérationnelle supplémentaire et les risques accrus en termes de disponibilité et de performances dépasseront les avantages en termes de sécurité perçus pour les magasins de clés externes.

En savoir plus :

- Announcing AWS KMS External Key Store (Annonce du magasin de clés externe) sur le Blog AWS News.

Ai-je besoin d'un magasin de clés externe ?

Pour la plupart des utilisateurs, le magasin de clés AWS KMS par défaut, qui est protégé par les modules de sécurité matériels validés FIPS 140-2, Niveau de sécurité 3, répond à leurs exigences en matière de sécurité, de contrôle et de réglementation. Les utilisateurs de magasins de clés externes supportent des coûts, une maintenance et une charge de dépannage considérables, ainsi que des risques en matière de latence, de disponibilité et de fiabilité.

Lorsque vous envisagez de choisir un magasin de clés externe, prenez le temps de comprendre les alternatives, notamment un magasin de clés AWS CloudHSM soutenu par un cluster AWS CloudHSM que vous possédez et gérez, et des clés KMS contenant des éléments de clé importés que vous générez dans vos propres modules HSM et que vous pouvez supprimer des clés KMS à

la demande. En particulier, l'importation d'éléments de clé ayant un délai d'expiration très court peut fournir un niveau de contrôle similaire sans risques pour la performance ou la disponibilité.

Un magasin de clés externe peut être la solution adaptée à votre organisation si vous répondez aux exigences suivantes :

- Vous devez utiliser des clés cryptographiques dans votre gestionnaire de clés sur site ou dans un gestionnaire de clés en dehors d'AWS que vous contrôlez.
- Vous devez prouver que vos clés cryptographiques sont conservées uniquement sous votre contrôle en dehors du cloud.
- Vous devez pouvoir chiffrer et déchiffrer à l'aide de clés cryptographiques disposant d'une autorisation indépendante.
- Les clés doivent être soumises à un chemin d'audit indépendant secondaire.

Si vous choisissez un magasin de clés externe, limitez son utilisation aux charges de travail qui nécessitent une protection au moyen de clés cryptographiques en dehors d'AWS.

Modèle de responsabilité partagée

Les clés KMS standard utilisent des éléments de clé générés et utilisés dans les modules HSM qu'AWS KMS possède et gère. Vous établissez les politiques de contrôle d'accès sur vos clés KMS et configurez les Services AWS qui utilisent les clés KMS pour protéger vos ressources. AWS KMS endosse la responsabilité en matière de sécurité, de disponibilité, de latence et de durabilité des éléments de clé de vos clés KMS.

Les clés KMS des magasins de clés externes dépendent des éléments et des opérations de clé de votre gestionnaire de clés externe. À ce titre, l'équilibre des responsabilités penche en votre faveur. Vous êtes responsable de la sécurité, de la fiabilité, de la durabilité et des performances des clés cryptographiques de votre gestionnaire de clés externe. AWS KMS est chargé de répondre rapidement aux requêtes et de communiquer avec le proxy de votre magasin de clés externe, ainsi que de maintenir nos normes de sécurité. Pour garantir que chaque clé externe stocke un texte chiffré au moins aussi robuste que le texte chiffré AWS KMS standard, AWS KMS chiffre d'abord tout le texte brut avec des éléments de clé AWS KMS spécifiques à votre clé KMS, puis l'envoie à votre gestionnaire de clés externe pour chiffrement à l'aide de votre clé externe, une procédure connue sous le nom de double chiffrement. Par conséquent, ni AWS KMS ni le propriétaire des éléments de clé externes ne peuvent déchiffrer seuls le texte chiffré à double chiffrement.

Vous êtes responsable du maintien d'un gestionnaire de clés externe qui répond à vos normes réglementaires et de performance, de la fourniture et de la maintenance d'un proxy de magasin de clés externe conforme à la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie), ainsi que de la disponibilité et de la durabilité de vos éléments de clé. Vous devez également créer, configurer et maintenir un magasin de clés externe. Lorsque des erreurs surviennent et sont causées par des composants que vous gérez, vous devez être prêt à les identifier et à les résoudre afin que les services AWS puissent accéder à vos ressources sans interruption excessive. AWS KMS fournit des conseils de dépannage pour vous aider à déterminer la cause des problèmes et les résolutions les plus probables.

Consultez les CloudWatch statistiques et les dimensions Amazon AWS KMS enregistrées pour les principaux magasins externes. AWS KMSrecommande vivement de créer des CloudWatch alarmes pour surveiller votre magasin de clés externe afin de détecter les premiers signes de performances et de problèmes opérationnels avant qu'ils ne surviennent.

Qu'est-ce qui change ?

Les magasins de clés externes ne prennent en charge que les clés KMS de chiffrement symétriques. Au sein d'AWS KMS, vous utilisez et gérez les clés KMS dans un magasin de clés externe de la même manière que vous gérez les autres clés gérées par le client, notamment en définissant des politiques de contrôle d'accès et en surveillant l'utilisation des clés. Vous utilisez les mêmes API avec les mêmes paramètres pour solliciter une opération cryptographique avec une clé KMS dans un magasin de clés externe que celles que vous utilisez pour toute clé KMS. La tarification est également la même que pour les clés KMS standard. Pour plus d'informations, veuillez consulter la rubrique Gérer des clés KMS dans un magasin de clés externe, Utiliser des clés KMS dans un magasin de clés externe, ainsi que la Tarification AWS Key Management Service.

Toutefois, avec les magasins de clés externes, les principes suivants changent :

- Vous êtes responsable de la disponibilité, de la durabilité et de la latence des opérations de clé.
- Vous êtes responsable de tous les coûts liés au développement, à l'achat, à l'exploitation et à la licence de votre système de gestion de clés externe.
- Vous pouvez implémenter une autorisation indépendante pour toutes les requêtes provenant d'AWS KMS à destination de votre proxy de magasin de clés externe.
- Vous pouvez surveiller, auditer et journaliser toutes les opérations de votre proxy de magasin de clés externe, ainsi que toutes les opérations de votre gestionnaire de clés externe liées aux requêtes AWS KMS.

Où commencer ?

Pour créer et gérer un magasin de clés externe, vous devez choisir l'option de connectivité du proxy de votre magasin de clés externe, réunir les conditions préalables, puis créer et configurer votre magasin de clés externe. Pour commencer, veuillez consulter la rubrique Planifier un magasin de clés externe.

Quotas

AWS KMS autorise jusqu'à 10 magasins de clés personnalisés dans chaque Compte AWS et région, y compris les magasins de clés AWS CloudHSM et les magasins de clés externes, quel que soit leur état de connexion. En outre, il existe des quotas de requêtes AWS KMS concernant l'utilisation des clés KMS dans un magasin de clés externes.

Si vous choisissez la connectivité proxy VPC pour votre proxy de magasin de clés externe, des quotas peuvent également s'appliquer aux composants requis, tels que les VPC, les sous-réseaux et les équilibreurs de charge réseau. Pour plus d'informations sur ces quotas, consultez la console Service Quotas.

Régions

Pour minimiser la latence du réseau, créez les composants de votre magasin de clés externe dans la Région AWS la plus proche de votre gestionnaire de clés externe. Si possible, choisissez une région dont le temps d'aller-retour sur le réseau (RTT, round-trip time) est inférieur ou égal à 35 millisecondes.

Les magasins de clés externes sont pris en charge dans tous Régions AWS dans lesquels AWS KMS est pris en charge à l'exception de la Chine (Beijing) et Chine (Ningxia).

Fonctions non prises en charge

AWS KMS ne prend pas en charge les fonctions suivantes dans les magasins de clés personnalisés.

- Clés KMS asymétriques
- Paires de clés de données asymétriques
- Clés KMS HMAC
- Clés KMS avec des éléments de clé importés

- Rotation automatique des clés
- Clés multi-région

Rubriques

- Concepts de magasins de clés externes
- Fonctionnement des magasins de clés externes
- Contrôler l'accès à votre magasin de clés externe
- Planifier un magasin de clés externe
- Gérer un magasin de clés externe
- Gérer des clés KMS dans un magasin de clés externe
- Résoudre les problèmes liés aux magasins de clés externes

Concepts de magasins de clés externes

Cette rubrique explique certains des concepts utilisés dans les magasins de clés externes.

Rubriques

- Magasin de clés externe
- Gestionnaire de clés externe
- Clé externe
- Proxy de magasin de clés externe
- Connectivité du proxy de magasin de clés externe
- Informations d'identification pour l'authentification du proxy de magasin de clés externe
- API de proxy
- Double chiffrement

Magasin de clés externe

Un magasin de clés externe est un magasin de clés personnalisé AWS KMS soutenu par un gestionnaire de clés externe en dehors d'AWS que vous possédez et gérez. Chaque clé KMS d'un magasin de clés externe est associée à une clé externe dans votre gestionnaire de clés externe. Lorsque vous utilisez une clé KMS dans un magasin de clés externe à des fins de chiffrement ou de déchiffrement, l'opération est exécutée dans votre gestionnaire de clés externe à l'aide de votre clé

externe, une disposition connue sous le nom de Hold your Own Keys (HYOK). Cette fonctionnalité est conçue pour les entreprises qui sont tenues de conserver leurs clés cryptographiques dans leur propre gestionnaire de clés externe.

Les magasins de clés externes garantissent que les clés et les opérations cryptographiques qui protègent vos ressources AWS restent dans votre gestionnaire de clés externe sous votre contrôle. AWS KMS envoie des requêtes à votre gestionnaire de clés externe pour chiffrer et déchiffrer des données, mais AWS KMS ne peut pas créer, supprimer ou gérer de clés externes. Toutes les requêtes provenant d'AWS KMS à destination de votre gestionnaire de clés externe sont transmises par un composant logiciel de proxy de magasin de clés externe que vous fournissez, possédez et gérez.

Les services AWS qui prennent en charge les clés AWS KMS gérées par le client peuvent utiliser les clés KMS qui se trouvent dans votre magasin de clés externe pour protéger vos données. En définitive, vos données sont bel et bien protégées par vos clés à l'aide de vos opérations de chiffrement dans votre gestionnaire de clés externe.

Les clés KMS d'un magasin de clés externe présentent des modèles de confiance, des accords de responsabilité partagée et des attentes en matière de performances fondamentalement différents de ceux des clés KMS standard. Avec les magasins de clés externes, vous êtes responsable de la sécurité et de l'intégrité des éléments de clé et des opérations cryptographiques. La disponibilité et la latence des clés KMS dans un magasin de clés externe sont affectées par le matériel, les logiciels, les composants réseau ainsi que par la distance entre AWS KMS et votre gestionnaire de clés externe. Vous êtes également susceptible d'encourir des coûts supplémentaires pour votre gestionnaire de clés externe et pour l'infrastructure de mise en réseau et d'équilibrage de charge dont vous avez besoin pour que votre gestionnaire de clés externe communique avec AWS KMS.

Vous pouvez utiliser votre magasin de clés externe dans le cadre de votre stratégie globale de protection des données. Pour chaque ressource AWS que vous protégez, vous pouvez décider laquelle nécessite une clé KMS dans un magasin de clés externe et laquelle peut être protégée par une clé KMS standard. Cela vous donne la possibilité de choisir des clés KMS pour des classifications de données, des applications ou des projets spécifiques.

Gestionnaire de clés externe

Un gestionnaire de clés externe est un composant extérieur à AWS qui peut générer des clés symétriques AES 256 bits et effectuer un chiffrement et un déchiffrement symétriques. Le gestionnaire de clés externe pour un magasin de clés externe peut être un module de sécurité matérielle (HSM) physique, un module HSM virtuel ou un gestionnaire de clés logiciel avec ou sans

composant HSM. Il peut être situé n'importe où à l'extérieur d'AWS, y compris dans vos locaux, dans un centre de données sur site ou à distance, ou dans n'importe quel cloud. Votre magasin de clés externe peut être soutenu par un seul gestionnaire de clés externe ou par plusieurs instances de gestionnaire de clés connexes qui partagent des clés cryptographiques, comme un cluster HSM. Les magasins de clés externes sont conçus pour prendre en charge divers gestionnaires externes provenant de différents fournisseurs. Pour plus d'informations sur les exigences relatives à votre gestionnaire de clés externe, veuillez consulter la rubrique Planifier un magasin de clés externe.

Clé externe

Chaque clé KMS d'un magasin de clés externe est associée à une clé cryptographique dans votre gestionnaire de clés externe appelée clé externe. Lorsque vous chiffrez ou déchiffrez avec une clé KMS dans votre magasin de clés externe, l'opération cryptographique est effectuée dans votre gestionnaire de clés externe à l'aide de votre clé externe.

> **Warning**
>
> La clé externe est essentielle au fonctionnement de la clé KMS. En cas de perte ou de suppression de la clé externe, le texte chiffré au moyen de la clé KMS associée est irrécupérable.

Pour les magasins de clés externes, une clé externe doit être une clé AES 256 bits activée et capable d'effectuer le chiffrement et le déchiffrement. Pour obtenir des informations détaillées sur les exigences relatives aux clés externes, veuillez consulter la rubrique Exigences relatives à une clé KMS dans un magasin de clés externe.

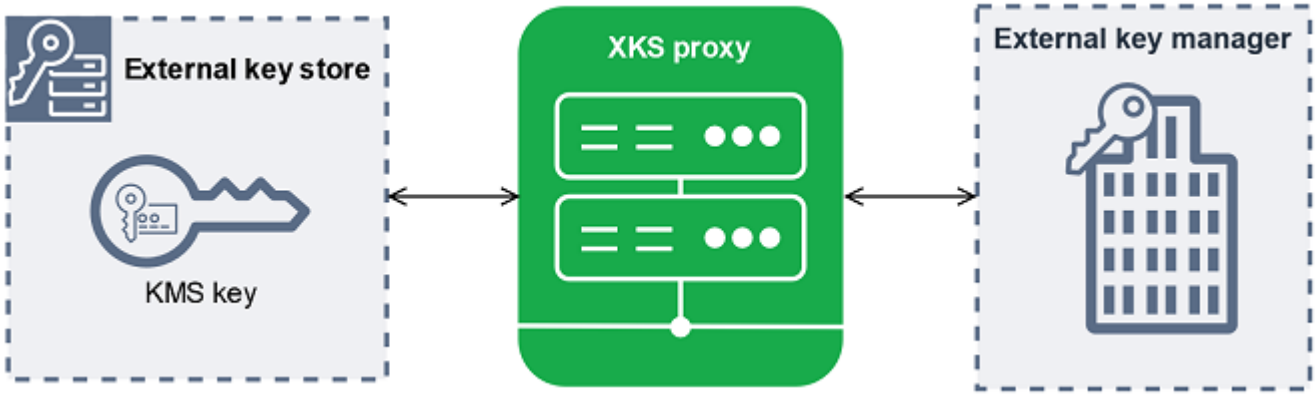
AWS KMS ne peut pas créer, supprimer ou gérer des clés externes. Les éléments de votre clé cryptographique ne quittent jamais votre gestionnaire de clés externe. Lorsque vous créez une clé KMS dans un magasin de clés externe, vous fournissez l'ID d'une clé externe (`XksKeyId`). Vous ne pouvez pas modifier l'ID de clé externe associé à une clé KMS, bien que votre gestionnaire de clés externe puisse effectuer une rotation des éléments de clé associés à l'ID de clé externe.

Outre votre clé externe, une clé KMS contenue dans un magasin de clés externe contient également des éléments de clé AWS KMS. Les données protégées par la clé KMS sont chiffrées d'abord par AWS KMS à l'aide des éléments de clé AWS KMS, puis par votre gestionnaire de clés externe à l'aide de votre clé externe. Ce processus de double chiffrement garantit que le texte chiffré protégé par votre clé KMS est toujours au moins aussi robuste que le texte chiffré protégé uniquement par AWS KMS.

De nombreuses clés cryptographiques possèdent différents types d'identifiants. Lorsque vous créez une clé KMS dans un magasin de clés externe, indiquez l'ID de la clé externe que le proxy de magasin de clés externe utilise pour faire référence à la clé externe. Si vous utilisez le mauvais identifiant, votre tentative de créer une clé KMS dans votre magasin de clés externe échoue.

Proxy de magasin de clés externe

Le proxy de magasin de clés externe (« proxy XKS ») est une application logicielle détenue et gérée par le client qui assure la médiation de toutes les communications entre AWS KMS et votre gestionnaire de clés externe. Il traduit également les requêtes AWS KMS génériques dans un format que le gestionnaire de clés externe propre à votre fournisseur comprend. Un proxy de magasin de clés externe est requis pour un magasin de clés externe. Chaque magasin de clés externe est associé à exactement un proxy de magasin de clés externe.



AWS KMS ne peut pas créer, supprimer ou gérer des clés externes. Vos éléments de clé cryptographique ne quittent jamais votre gestionnaire de clés externe. Toute communication entre AWS KMS et votre gestionnaire de clés externe est assurée par votre proxy de magasin de clés externe. AWS KMS envoie des requêtes au proxy de magasin de clés externe et reçoit des réponses du proxy de magasin de clés externe. Le proxy de magasin de clés externe est chargé de transmettre les requêtes d'AWS KMS à votre gestionnaire de clés externe et de renvoyer les réponses de votre gestionnaire de clés externe à AWS KMS.

Vous possédez et gérez le proxy de magasin de clés externe pour votre magasin de clés externe, et vous êtes responsable de sa maintenance et de son fonctionnement. Vous pouvez développer votre proxy de magasin de clés externe en vous basant sur la spécification de l'API du proxy de magasin de clés externe (langue française non garantie) open source qu'AWS KMS publie ou acheter une application proxy auprès d'un fournisseur. Votre proxy de magasin de clés externe est peut-être inclus dans votre gestionnaire de clés externe. Pour faciliter le développement de proxy, fournit AWS KMS également un exemple de proxy de stockage de clés externe (aws-kms-xks-proxy) et un client

de test (xks-kms-xksproxy-test-client) qui vérifie que votre proxy de stockage de clés externe est conforme à la spécification.

Pour s'authentifier sur AWS KMS, le proxy utilise des certificats TLS côté serveur. Pour vous authentifier auprès de votre proxy, AWS KMS signe toutes les requêtes adressées à votre proxy de magasin de clés externe à l'aide d'informations d'identification pour l'authentification du proxy SigV4. Votre proxy peut éventuellement activer le protocole TLS mutuel (mTLS) afin de garantir qu'il n'accepte que les requêtes émises par AWS KMS.

Votre proxy de stockage de clés externe doit prendre en charge HTTP/1.1 ou version ultérieure et TLS 1.2 ou version ultérieure avec au moins l'une des suites de chiffrement suivantes :

- TLS_AES_256_GCM_SHA384 (TLS 1.3)
- TLS_CHACHA20_POLY1305_SHA256 (TLS 1.3)

> **Note**
>
> Le AWS GovCloud (US) Region ne prend pas en charge TLS_CHACHA20_POLY1305_SHA256.

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)

Pour créer et utiliser les clés KMS dans votre magasin de clés externe, vous devez d'abord connecter le magasin de clés externe à son proxy de magasin de clés externe. Vous pouvez également déconnecter votre magasin de clés externe de son proxy à la demande. Dans ce cas, toutes les clés KMS du magasin de clés externe deviennent indisponibles ; elles ne peuvent être utilisées dans aucune opération cryptographique.

Connectivité du proxy de magasin de clés externe

La connectivité du proxy de magasin de clés externe (« connectivité proxy XKS ») décrit la méthode qu'AWS KMS utilise pour communiquer avec le proxy de magasin de clés externe.

Vous spécifiez votre option de connectivité de proxy lorsque vous créez votre magasin de clés externe, et elle devient une propriété du magasin de clés externe. Vous pouvez modifier l'option de connectivité de votre proxy en mettant à jour la propriété du magasin de clés personnalisé, mais vous devez vous assurer que votre proxy de magasin de clés externe peut toujours accéder aux mêmes clés externes.

AWS KMS prend en charge les options de connectivité suivantes :

- Connectivité au point de terminaison public : AWS KMS envoie des requêtes pour votre proxy de magasin de clés externe via Internet à un point de terminaison public que vous contrôlez. Cette option est simple à créer et à gérer, mais elle peut ne pas répondre aux exigences de sécurité pour chaque installation.
- Connectivité au service de point de terminaison d'un VPC : AWS KMS envoie des requêtes à un service de point de terminaison Amazon Virtual Private Cloud (Amazon VPC) que vous créez et gérez. Vous pouvez héberger votre proxy de magasin de clés externe au sein de votre Amazon VPC, ou héberger votre proxy de magasin de clés externe à l'extérieur d'AWS et utiliser Amazon VPC uniquement pour la communication.

Pour plus d'informations sur les options de connectivité du proxy de magasin de clés externe, veuillez consulter la rubrique Choisir une option de connectivité de proxy.

Informations d'identification pour l'authentification du proxy de magasin de clés externe

Pour vous authentifier auprès du proxy de votre magasin de clés externe, AWS KMS signe toutes les requêtes adressées au proxy de votre magasin de clés externe à l'aide d'informations d'identification d'authentification Signature V4 (SigV4). Vous établissez et gérez les informations d'identification de l'authentification sur votre proxy, puis vous les fournissez à AWS KMS lorsque vous créez votre magasin externe.

> **Note**
>
> Les informations d'identification SigV4 qu'AWS KMS utilise pour signer les requêtes adressées au proxy XKS ne sont pas liées aux informations d'identification SigV4 associées aux principaux de AWS Identity and Access Management dans vos Comptes AWS. Ne réutilisez aucune information d'identification IAM SigV4 pour votre proxy de magasin de clés externe.

Chaque information d'identification pour l'authentification du proxy comporte deux parties. Vous devez fournir les deux parties lors de la création d'un magasin de clés externe ou de la mise à jour des informations d'identification de l'authentification pour votre magasin de clés externe.

- ID de la clé d'accès : identifie la clé d'accès secrète. Vous pouvez fournir cet ID en texte brut.

- Clé d'accès secrète : partie secrète des informations d'identification. AWS KMS chiffre la clé d'accès secrète contenue dans les informations d'identification avant de les stocker.

Vous pouvez modifier le paramètre des informations d'identification à tout moment, par exemple lorsque vous saisissez des valeurs incorrectes, lorsque vous modifiez vos informations d'identification sur le proxy ou lorsque votre proxy effectue une rotation des informations d'identification. Pour plus de détails techniques sur l'authentification AWS KMS auprès du proxy de magasin de clés externe, veuillez consulter la rubrique Authentification dans la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie).

Pour vous permettre d'effectuer une rotation de vos informations d'identification sans perturber les Services AWS qui utilisent des clés KMS dans votre magasin de clés externe, nous recommandons que le proxy de magasin de clés externe prenne en charge au moins deux informations d'identification valides pour AWS KMS. Cela garantit que vos anciennes informations d'identification continuent de fonctionner pendant que vous fournissez vos nouvelles informations d'identification à AWS KMS.

Pour vous aider à suivre l'âge de votre identifiant d'authentification proxy, définissez AWS KMS une CloudWatch métrique Amazon, XksProxyCredentialAge. Vous pouvez utiliser cette métrique pour créer une CloudWatch alarme qui vous avertit lorsque l'âge de votre identifiant atteint un seuil que vous avez établi.

Pour garantir que votre proxy de magasin de clés externe ne réponde qu'à AWS KMS, certains proxys de clés externes prennent en charge le protocole Transport Layer Security mutuel (mTLS). Pour plus de détails, consultez Authentification mTLS (facultatif).

API de proxy

Pour prendre en charge un magasin de clés externe AWS KMS, un proxy de magasin de clés externe doit implémenter les API de proxy requises, comme décrit dans la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie). Ces requêtes API du proxy sont les seules qu'AWS KMS envoie au proxy. Bien que vous n'envoyiez jamais ces requêtes directement, le fait de les connaître peut vous aider à résoudre les problèmes qui pourraient survenir avec votre magasin de clés externe ou son proxy. Par exemple, AWS KMS inclut des informations sur la latence et les taux de réussite de ces appels d'API dans ses CloudWatch statistiques Amazon pour les magasins de clés externes. Pour plus de détails, consultez Surveiller un magasin de clés externe.

Le tableau suivant répertorie et décrit chacune des API de proxy. Il inclut également les opérations AWS KMS qui déclenchent un appel à l'API de proxy et toutes les exceptions d'opération AWS KMS liées à l'API de proxy.

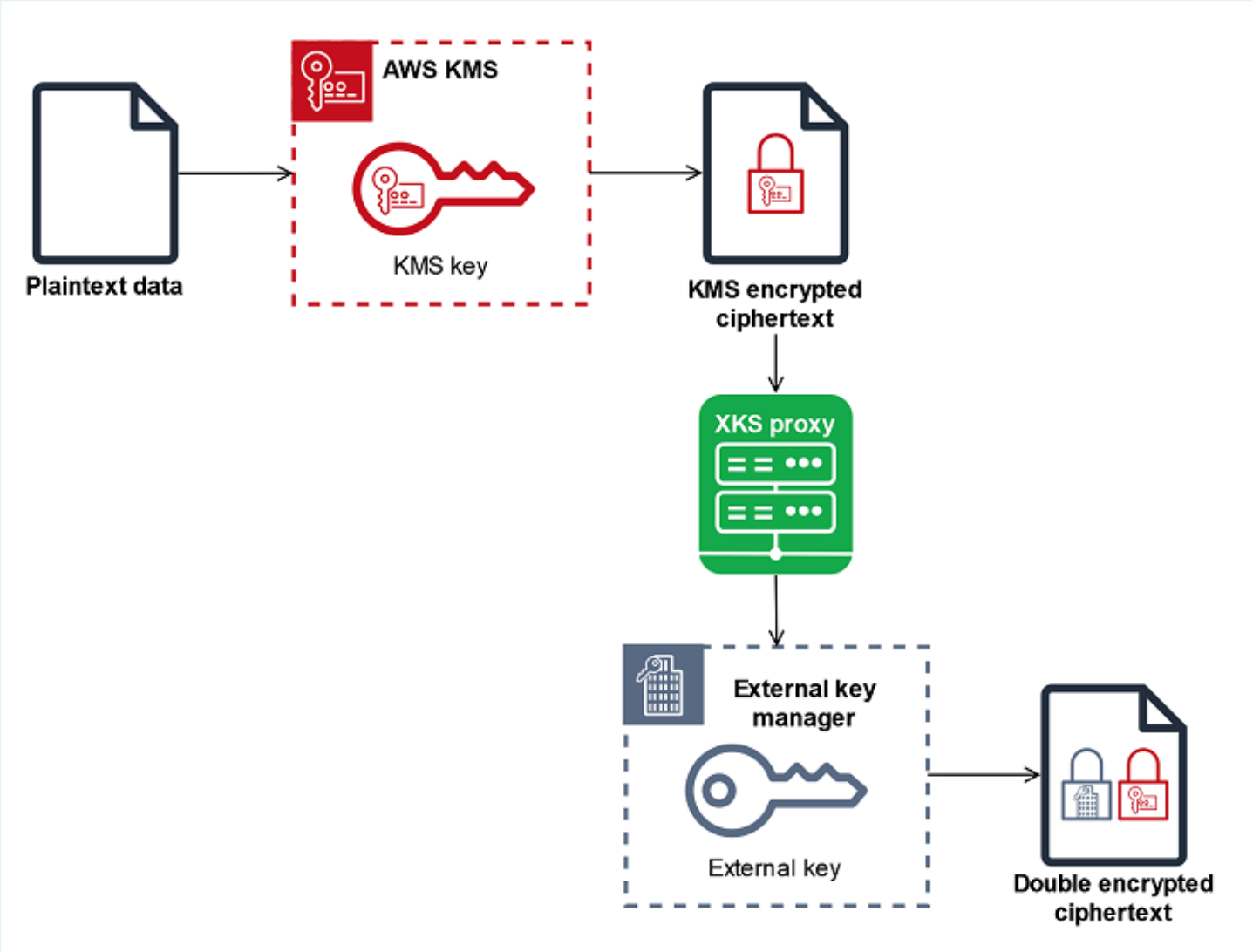
API de proxy	Description	Opérations AWS KMS liées
Decrypt	AWS KMS envoie le texte chiffré à déchiffrer et l'ID de la clé externe à utiliser. L'algorithme de chiffrement requis est AES_GCM.	Décrypter, ReEncrypt
Encrypt	AWS KMS envoie les données à chiffrer et l'ID de la clé externe à utiliser. L'algorithme de chiffrement requis est AES_GCM.	Chiffrer, GenerateDataKey, GenerateDataKeyWithoutPlaintextReEncrypt
GetHealthStatus	AWS KMS demande des informations sur l'état du proxy et de votre gestionnaire de clés externe. L'état de chaque gestionnaire de clés externe peut être l'un des suivants. • `Active` : sain ; peut assurer le trafic • `Degraded` : défectueux, mais peut assurer le trafic • `Unavailable` : défectueux ; ne peut pas assurer le trafic	CreateCustomKeyStore(pour la connectivité des points de terminaison publics), ConnectCustomKeyStore(pour la connectivité des services de point de terminaison VPC) Si toutes les instances externes du gestionnaire de clés sont `Unavailable`, les tentatives de création ou de connexion du magasin de clés échouent avec l'exception `XksProxyUriUnreachableException`.
GetKeyMetadata	AWS KMS demande des informations sur la clé externe associée à une clé KMS dans votre magasin de clés externe. La réponse inclut la spécification de la clé (`AES_256`), l'utilisation de la clé (`[ENCRYPT, DECRYPT]`) et	CreateKey Si la spécification de la clé n'est pas `AES_256`, si l'utilisation de la clé n'est pas `[ENCRYPT, DECRYPT]`, ou si l'état est `DISABLED`, l'opération `CreateKey` échoue avec

API de proxy	Description	Opérations AWS KMS liées
	indique si la clé externe est `ENABLED` ou `DISABLED`.	l'exception `XksKeyInvalidConfigurationException` .

Double chiffrement

Les données chiffrées par une clé KMS dans un magasin de clés externe sont chiffrées deux fois. Tout d'abord, AWS KMS chiffre les données avec des éléments de clé AWS KMS spécifiques à la clé KMS. Ensuite, le texte chiffré au moyen d'AWS KMS est chiffré par votre gestionnaire de clés externe à l'aide de votre clé externe. Ce processus est connu sous le nom de double chiffrement.

Le double chiffrement garantit que les données chiffrées par une clé KMS dans un magasin de clés externe sont au moins aussi robustes que le texte chiffré au moyen d'une clé KMS standard. Il protège également votre texte brut en transit depuis AWS KMS vers votre proxy de magasin de clés externe. Grâce au double chiffrement, vous gardez le contrôle total de vos textes chiffrés. Si vous révoquez définitivement l'accès d'AWS à votre clé externe par le biais de votre proxy externe, tout texte chiffré restant dans AWS est en fait détruit par chiffrement.



Pour activer le double chiffrement, chaque clé KMS d'un magasin de clés externe possède deux clés de sauvegarde cryptographiques :

- Des éléments de clé AWS KMS uniques à la clé KMS. Ces éléments de clé sont générés et utilisés uniquement dans les modules de sécurité matériels certifiés FIPS 140-2 - Niveau de sécurité 3 AWS KMS.
- Une clé externe dans votre gestionnaire de clés externe.

Le double chiffrement a les effets suivants :

- AWS KMS ne peut pas déchiffrer un texte chiffré au moyen d'une clé KMS dans un magasin de clés externe sans accéder à vos clés externes via votre proxy de magasin de clés externe.

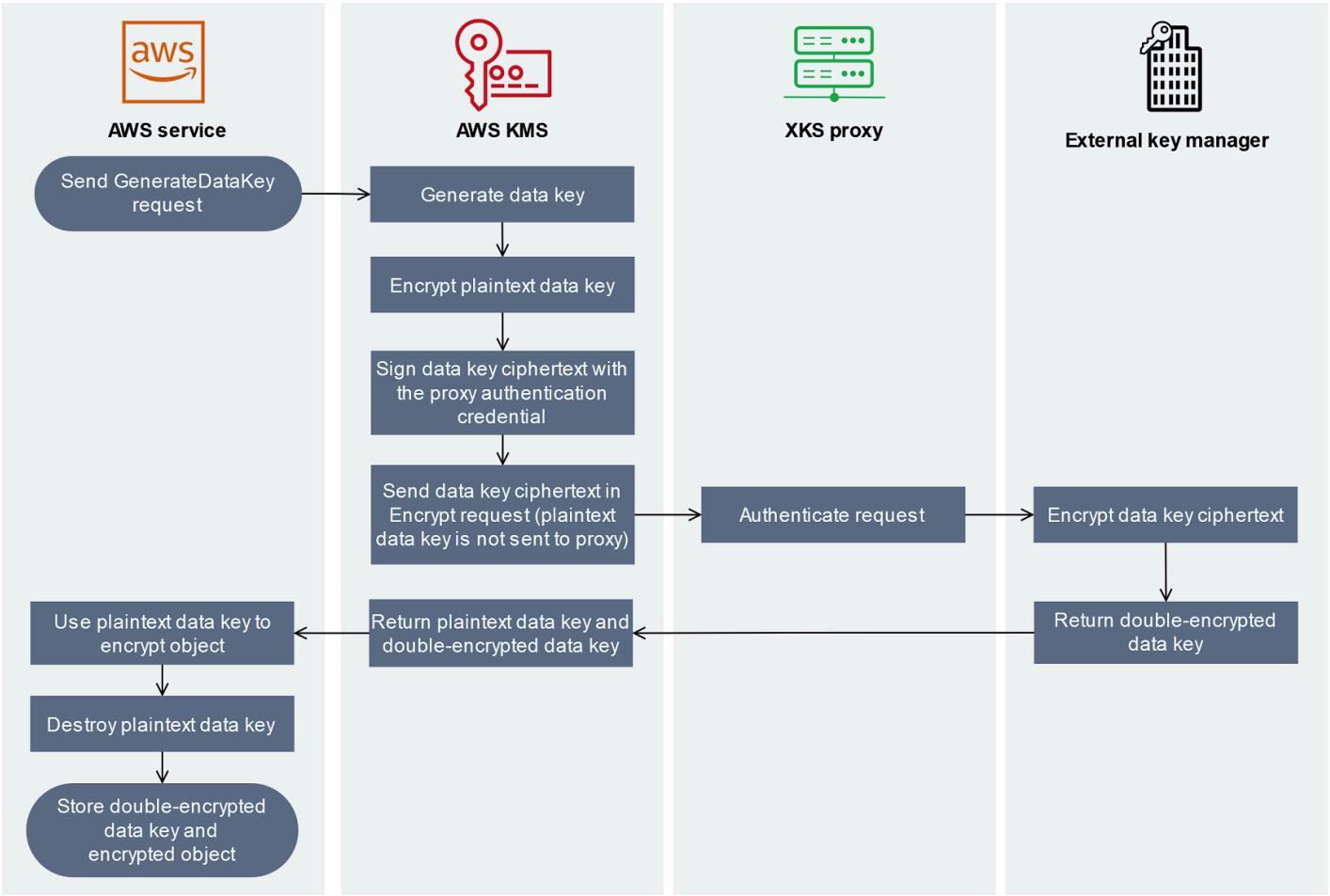
- Vous ne pouvez pas déchiffrer un texte chiffré au moyen d'une clé KMS dans un magasin de clés externe situé à l'extérieur d'AWS, même si vous possédez sa clé externe.
- Vous ne pouvez pas recréer une clé KMS qui a été supprimée d'un magasin de clés externe, même si vous possédez ses éléments de clé externes. Chaque clé KMS possède des métadonnées uniques qu'elle inclut dans le texte chiffré symétrique. Une nouvelle clé KMS ne serait pas en mesure de déchiffrer le texte chiffré au moyen de la clé d'origine, même si elle utilisait les mêmes éléments de clé externes.

Pour un exemple de double chiffrement en pratique, veuillez consulter la rubrique Fonctionnement des magasins de clés externes.

Fonctionnement des magasins de clés externes

Votre magasin de clés externe, votre proxy de magasin de clés externe et votre gestionnaire de clés externe travaillent ensemble pour protéger vos ressources AWS. La procédure suivante décrit le flux de travail de chiffrement d'un Service AWS typique qui chiffre chaque objet sous une clé de données unique protégée par une clé KMS. Dans ce cas, vous avez choisi une clé KMS dans un magasin de clés externe pour protéger l'objet. L'exemple montre comment AWS KMS utilise le double chiffrement pour protéger la clé de données en transit et garantir que le texte chiffré généré par une clé KMS dans un magasin de clés externe est toujours au moins aussi robuste que le texte chiffré qui a été chiffré par une clé KMS symétrique standard contenant des éléments de clé dans AWS KMS.

Les méthodes de chiffrement utilisées par chaque Service AWS réel qui s'intègre à AWS KMS varient. Pour plus de détails, veuillez consulter la rubrique « Protection des données » dans le chapitre Sécurité de la documentation Service AWS.



1. Vous ajoutez un nouvel objet à votre ressource Service AWS. Pour chiffrer l'objet, Service AWS envoie une GenerateDataKeydemande à l'AWS KMSaide d'une clé KMS dans votre banque de clés externe.
2. AWS KMS génère une clé de données symétrique de 256 bits et prépare l'envoi d'une copie de la clé de données en texte brut à votre gestionnaire de clés externe via votre proxy de magasin de clés externe. AWS KMS commence le processus de double chiffrement en chiffrant la clé de données en texte brut avec les éléments de clé AWS KMS associés à la clé KMS dans le magasin de clés externe.
3. AWS KMS envoie une requête de chiffrement au proxy de magasin de clés externe associé au magasin de clés externe. La requête inclut le texte chiffré de la clé de données à chiffrer et l'ID de la clé externe associée à la clé KMS. AWS KMS signe la requête à l'aide des informations d'identification d'authentification du proxy pour votre proxy de magasin de clés externe.

   La copie en texte brut de la clé de données n'est pas envoyée au proxy du magasin de clés externes.

4. Le proxy de magasin de clés externe authentifie la requête, puis transmet la requête de chiffrement à votre gestionnaire de clés externe.

   Certains proxys de magasin de clés externe implémentent également une politique d'autorisation facultative qui permet uniquement à certains principaux d'effectuer des opérations dans des conditions spécifiques.

5. Votre gestionnaire de clés externe chiffre le texte chiffré de la clé de données à l'aide de la clé externe spécifiée. Le gestionnaire de clés externe renvoie la clé de données doublement chiffrée à votre proxy de magasin de clés externe, qui la renvoie à AWS KMS.
6. AWS KMS renvoie à l'Service AWS la clé de données en texte brut et la copie doublement chiffrée de cette clé de données.
7. L'Service AWS utilise la clé de données en texte brut pour chiffrer l'objet ressource, détruit la clé de données en texte brut et stocke la clé de données chiffrée avec l'objet chiffré.

   Certains Services AWS peuvent mettre en cache la clé de données en texte brut afin de l'utiliser pour plusieurs objets ou de la réutiliser tant que la ressource est en service. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

Pour déchiffrer l'objet chiffré, l'Service AWS doit renvoyer la clé des données chiffrées à AWS KMS dans une requête Decrypt. Pour déchiffrer la clé de données chiffrée, AWS KMS doit renvoyer la clé de données chiffrée à votre proxy de magasin de clés externe avec l'ID de la clé externe. Si la requête de déchiffrement adressée au proxy de magasin de clés externe échoue pour quelque raison que ce soit, AWS KMS ne peut pas déchiffrer la clé de données chiffrée et l'Service AWS ne peut pas déchiffrer l'objet chiffré.

Contrôler l'accès à votre magasin de clés externe

Toutes les fonctionnalités de contrôle d'accès AWS KMS (politiques de clés, politiques IAM et octrois) que vous utilisez avec des clés KMS standard fonctionnent de la même manière pour les clés KMS d'un magasin de clés externe. Vous pouvez utiliser les politiques IAM pour contrôler l'accès aux opérations d'API qui créent et gèrent des magasins de clés externes. Vous pouvez utiliser les politiques IAM et les politiques de clés pour contrôler l'accès aux AWS KMS keys de votre magasin de clés externe. Vous pouvez également utiliser des politiques de contrôle des services pour votre organisation AWS et des politiques de point de terminaison d'un VPC pour contrôler l'accès aux clés KMS dans votre magasin de clés externe.

Nous vous recommandons de ne fournir aux utilisateurs et aux rôles que les autorisations dont ils ont besoin pour les tâches qu'ils sont susceptibles d'effectuer.

Rubriques

- Autoriser des gestionnaires de magasins de clés externes
- Autoriser les utilisateurs de clés KMS dans des magasins de clés externes
- Autoriser AWS KMS à communiquer avec le proxy de votre magasin de clés externe
- Autorisation par proxy de magasin de clés externe (facultatif)
- Authentification mTLS (facultatif)

Autoriser des gestionnaires de magasins de clés externes

Les principaux qui créent et gèrent un magasin de clés externe doivent être autorisés pour les opérations de magasin de clés personnalisé. La liste suivante décrit les autorisations minimales requises pour les gestionnaires de magasin de clés externe. Étant donné qu'un magasin de clés personnalisé n'est pas une ressource AWS, vous ne pouvez pas autoriser l'accès à un magasin de clés externe aux principaux d'autres Comptes AWS.

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms:DeleteCustomKeyStore`

Les principaux qui créent un magasin de clés externe doivent être autorisés à créer et à configurer les composants du magasin de clés externe. Les principaux ne peuvent créer des magasins de clés externes que sur leurs propres comptes. Pour créer un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC, les gestionnaires doivent être autorisés à créer les composants suivants :

- Un Amazon VPC
- Sous-réseaux publics et privés
- Un équilibreur de charge réseau et un groupe cible

- Un service de point de terminaison d'un Amazon VPC

Pour plus de détails, veuillez consulter les rubriques Identity and Access Management pour Amazon VPC, Identity and Access Management pour les points de terminaison de VPC et les services de points de terminaison de VPC et Elastic Load Balancing API permissions (Autorisations de l'API Elastic Load Balancing).

Autoriser les utilisateurs de clés KMS dans des magasins de clés externes

Les principaux qui créent et gèrent les AWS KMS keys de votre magasin de clés externe nécessitent les mêmes autorisations que ceux qui créent et gèrent les clés KMS dans AWS KMS. La politique de clé par défaut pour les clés KMS d'un magasin de clés externe est identique à la politique de clé par défaut pour les clés KMS dans AWS KMS. Le contrôle d'accès par attributs (ABAC), qui utilise des balises et des alias pour contrôler l'accès aux clés KMS, fonctionne également sur les clés KMS dans les magasins de clés externes.

Les principaux qui utilisent les clés KMS dans votre magasin de clés personnalisé pour les opérations de chiffrement ont besoin des autorisations pour effectuer les opérations de chiffrement avec la clé KMS, telle que kms:Decrypt. Vous pouvez fournir ces autorisations dans une politique IAM ou une politique de clé. Cependant, les principaux n'ont pas besoin d'autorisations supplémentaires pour utiliser une clé KMS dans un magasin de clés personnalisé.

Pour définir une autorisation qui s'applique uniquement aux clés KMS d'un magasin de clés externe, utilisez la condition de politique `kms:KeyOrigin` avec la valeur de `EXTERNAL_KEY_STORE`. Vous pouvez utiliser cette condition pour limiter l'CreateKeyautorisation kms : ou toute autorisation spécifique à une ressource clé KMS. Par exemple, la politique IAM suivante permet à l'identité à laquelle elle est associée d'appeler les opérations spécifiées sur toutes les clés KMS du compte, à condition que les clés KMS se trouvent dans un magasin de clés externe. Notez que vous pouvez limiter l'autorisation aux clés KMS dans un magasin de clés externe et aux clés KMS sur un Compte AWS, mais pas à un magasin de clés externe particulier sur le compte.

```
{
  "Sid": "AllowKeysInExternalKeyStores",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
```

```
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL_KEY_STORE"
    }
  }
}
```

Autoriser AWS KMS à communiquer avec le proxy de votre magasin de clés externe

AWS KMS communique avec votre gestionnaire de clés externe uniquement via le proxy de magasin de clés externe que vous fournissez. AWS KMS s'authentifie auprès de votre proxy en signant ses requêtes à l'aide du processus Signature Version 4 (SigV4) en utilisant les informations d'identification pour l'authentification du proxy de magasin de clés externe que vous spécifiez. Si vous utilisez la connectivité au point de terminaison public pour votre proxy de magasin de clés externe, AWS KMS ne requiert aucune autorisation supplémentaire.

Toutefois, si vous utilisez la connectivité au service de point de terminaison d'un VPC, vous devez autoriser AWS KMS à créer un point de terminaison d'interface pour votre service de point de terminaison d'un Amazon VPC. Cette autorisation est requise, indépendamment du fait que le proxy de magasin de clés externe soit dans votre VPC ou qu'il soit situé ailleurs, mais qu'il utilise le service de point de terminaison d'un VPC pour communiquer avec AWS KMS.

AWS KMSPour autoriser la création d'un point de terminaison d'interface, utilisez la console Amazon VPC ou l'ModifyVpcEndpointServicePermissionsopération. Accordez des autorisations au principal suivant : `cks.kms.<region>.amazonaws.com`.

Par exemple, la commande AWS CLI suivante permet à AWS KMS de se connecter au service de point de terminaison d'un VPC spécifié dans la région USA Ouest (Oregon) (us-west-2). Avant d'utiliser cette commande, remplacez l'ID de service d'Amazon VPC et l'Région AWS par des valeurs valides pour votre configuration.

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

Pour supprimer cette autorisation, utilisez la console Amazon VPC ou ModifyVpcEndpointServicePermissionsavec le `RemoveAllowedPrincipals` paramètre.

Autorisation par proxy de magasin de clés externe (facultatif)

Certains proxys de magasin de clés externe mettent en œuvre des exigences d'autorisation pour l'utilisation de leurs clés externes. Un proxy de magasin de clés externe est autorisé, mais pas tenu, de concevoir et d'implémenter un schéma d'autorisation qui permet à des utilisateurs particuliers de demander des opérations particulières uniquement sous certaines conditions. Par exemple, un proxy peut être configuré pour permettre à l'utilisateur A de chiffrer avec une clé externe particulière, mais pas de déchiffrer à l'aide de cette clé.

L'autorisation par proxy est indépendante de l'authentification du proxy basée sur SigV4 qu'AWS KMS exige pour tous les proxys de magasin de clés externe. Elle est également indépendante des politiques de clés, des politiques IAM et des octrois qui accordent l'accès aux opérations affectant le magasin de clés externe ou ses clés KMS.

Pour activer l'autorisation par le proxy de magasin de clés externe, AWS KMS inclut des métadonnées dans chaque requête d'API de proxy, notamment l'appelant, la clé KMS, l'opération AWS KMS, l'Service AWS (le cas échéant). Les métadonnées de la requête pour la version 1 (v1) de l'API de proxy de clé externe se présentent comme suit.

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

Par exemple, vous pouvez configurer votre proxy pour autoriser les requêtes provenant d'un principal (`awsPrincipalArn`) particulier, mais uniquement lorsque la requête est faite au nom du principal par un Service AWS particulier (`kmsViaService`).

Si l'autorisation par proxy échoue, l'opération AWS KMS correspondante échoue avec un message expliquant l'erreur. Pour plus de détails, veuillez consulter la rubrique Problèmes d'autorisation du proxy.

Authentification mTLS (facultatif)

Pour permettre à votre proxy de magasin de clés externe d'authentifier les requêtes provenant de AWS KMS, AWS KMS signe toutes les requêtes adressées à votre proxy de magasin de clés externe

à l'aide des informations d'identification pour l'authentification du proxy Signature V4 (SigV4) pour votre magasin de clés externe.

Pour garantir davantage que votre proxy de magasin de clés externe ne réponde qu'aux requêtes AWS KMS, certains proxys de clé externes prennent en charge le protocole Transport Layer Security mutuel (mTLS), dans lequel les deux parties d'une transaction utilisent des certificats pour s'authentifier mutuellement. mTLS ajoute l'authentification côté client, dans laquelle le serveur proxy de magasin de clés externe authentifie le client AWS KMS, à l'authentification côté serveur fournie par le protocole TLS standard. Dans les rares cas où les informations d'identification d'authentification de votre proxy sont compromises, mTLS empêche une tierce partie d'effectuer des requêtes d'API au proxy de magasin de clés externe.

Pour implémenter le protocole mTLS, configurez votre proxy de magasin de clés externe de manière à n'accepter que les certificats TLS côté client présentant les propriétés suivantes :

- Le nom commun du sujet sur le certificat TLS doit être `cks.kms.<Region>.amazonaws.com`, par exemple, `cks.kms.eu-west-3.amazonaws.com`.
- Le certificat doit être lié à une autorité de certification associée à Amazon Trust Services.

Planifier un magasin de clés externe

Avant de créer votre magasin de clés externe, choisissez l'option de connectivité qui détermine comment AWS KMS communique avec les composants de votre magasin de clés externe. L'option de connectivité que vous choisissez détermine le reste du processus de planification.

En savoir plus :

- Passez en revue le processus de création d'un magasin de clés externe, ce qui inclut de réunir les conditions préalables. Cela vous aidera à vous assurer que vous disposez de tous les composants dont vous avez besoin lorsque vous créez votre magasin de clés externe.
- Découvrez comment contrôler l'accès à votre magasin de clés externe, notamment les autorisations requises par les administrateurs et les utilisateurs du magasin de clés externe.
- Découvrez les CloudWatch statistiques et les dimensions Amazon AWS KMS enregistrées pour les principaux magasins externes. Nous vous recommandons vivement de créer des alertes pour surveiller votre magasin de clés externe afin de détecter les premiers signes de problèmes de performance et de fonctionnement.

Choisir une option de connectivité de proxy

Si vous créez un magasin de clés externe, vous devez déterminer comment AWS KMS communique avec votre proxy de magasin de clés externe. Ce choix déterminera les composants dont vous avez besoin et la manière dont vous les configurez. AWS KMS prend en charge les options de connectivité suivantes. Choisissez l'option qui répond à vos objectifs de performance et de sécurité.

Avant de commencer, vérifiez que vous avez besoin d'un magasin de clés externe. La plupart des clients peuvent utiliser des clés KMS soutenues par des éléments de clé AWS KMS.

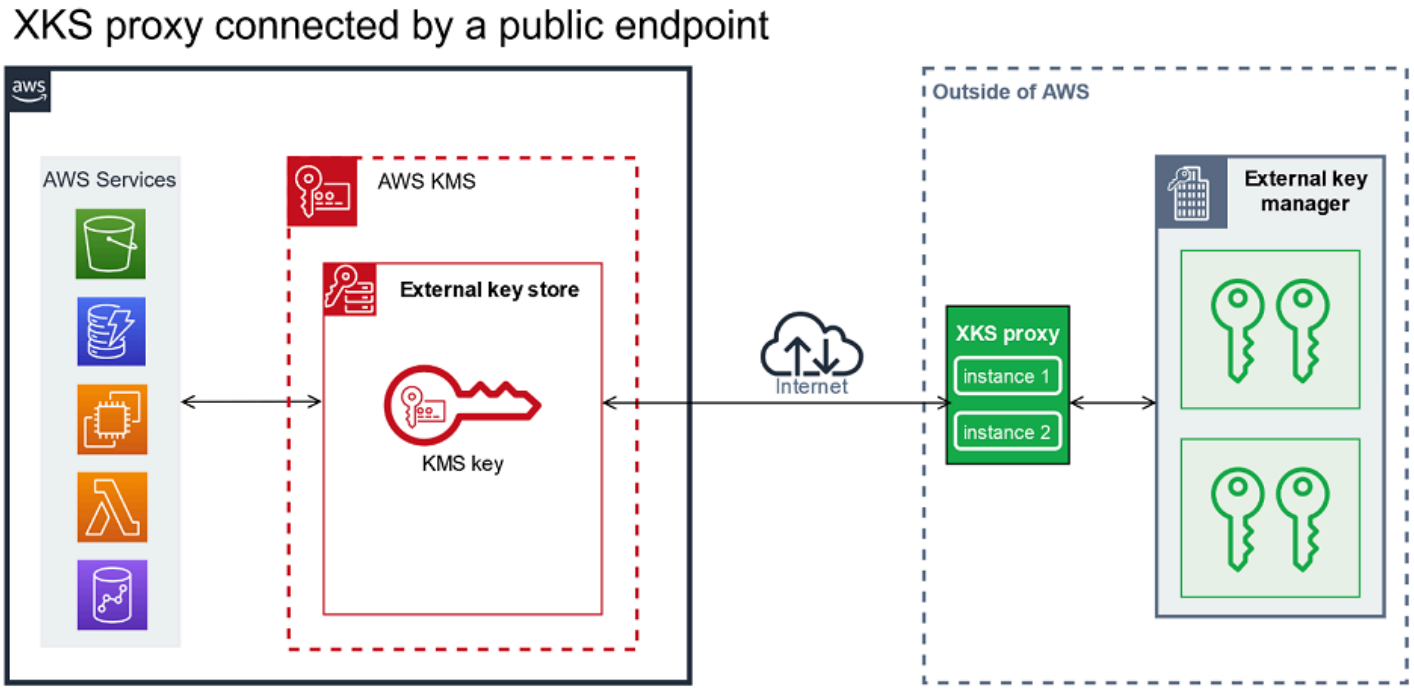
> **Note**
>
> Si votre proxy de magasin de clés externe est intégré à votre gestionnaire de clés externe, votre connectivité peut être prédéterminée. Pour obtenir des conseils, consultez la documentation de votre gestionnaire de clés externe ou de votre proxy de magasin de clés externe.

Vous pouvez modifier l'option de connectivité de votre proxy de magasin de clés externe, même sur un magasin de clés externe opérationnel. Toutefois, le processus doit être soigneusement planifié et exécuté afin de minimiser les interruptions, d'éviter les erreurs et de garantir un accès continu aux clés cryptographiques qui chiffrent vos données.

Connectivité au point de terminaison public

AWS KMS se connecte au proxy de magasin de clés externe (proxy XKS) via Internet à l'aide d'un point de terminaison public.

Cette option de connectivité est plus facile à configurer et à gérer, et elle s'adapte parfaitement à certains modèles de gestion des clés. Toutefois, il se peut qu'elle ne réponde pas aux exigences de sécurité de certaines organisations.



Prérequis

Si vous optez pour la connectivité au point de terminaison public, les éléments suivants sont requis.

- Le proxy de votre magasin de clés externe doit être accessible à partir d'un point de terminaison publiquement routable.
- Vous pouvez utiliser le même point de terminaison public pour plusieurs magasins de clés externes à condition qu'ils utilisent des valeurs de chemin d'URI de proxy différentes.
- Vous ne pouvez pas utiliser le même point de terminaison pour un magasin de clés externe doté d'une connectivité au point de terminaison public et pour un magasin de clés externe doté d'une connectivité aux services de point de terminaison d'un VPC dans la même Région AWS, même si les magasins de clés se trouvent dans des Comptes AWS différents.
- Vous devez obtenir un certificat TLS émis par une autorité de certification publique prise en charge pour les magasins de clés externes. Pour obtenir une liste, veuillez consulter les Autorités de certification approuvées (langue française non garantie).

  Le nom commun (CN) du sujet figurant sur le certificat TLS doit correspondre au nom de domaine indiqué dans le point de terminaison URI de proxy pour le proxy du magasin de clés externe. Par exemple, si le point de terminaison public est `https://myproxy.xks.example.com`, le TLS, le CN du certificat TLS doit être `myproxy.xks.example.com` ou `*.xks.example.com`.

- Assurez-vous que tous les pare-feux situés entre AWS KMS et le proxy de magasin de clés externe autorisent le trafic en provenance et à destination du port 443 sur le proxy. AWS KMS communique sur le port 443. Cette valeur n'est pas configurable.

Pour connaître toutes les exigences relatives à un magasin de clés externe, veuillez consulter la rubrique Réunir les conditions préalables.

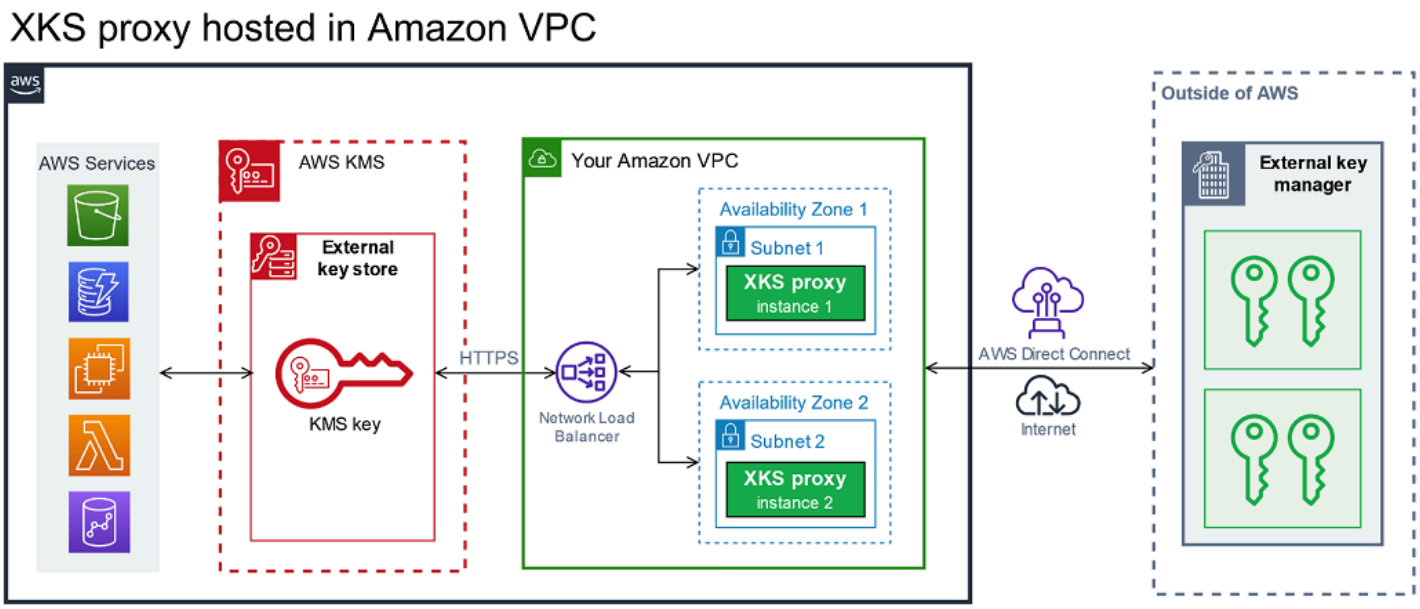
Connectivité au service de point de terminaison d'un VPC

AWS KMS se connecte au proxy de magasin de clés externe (proxy XKS) en créant un point de terminaison d'interface vers un service de point de terminaison d'un Amazon VPC que vous créez et configurez. Vous êtes responsable de la création du service de point de terminaison d'un VPC et de la connexion de votre VPC à votre gestionnaire de clés externe.

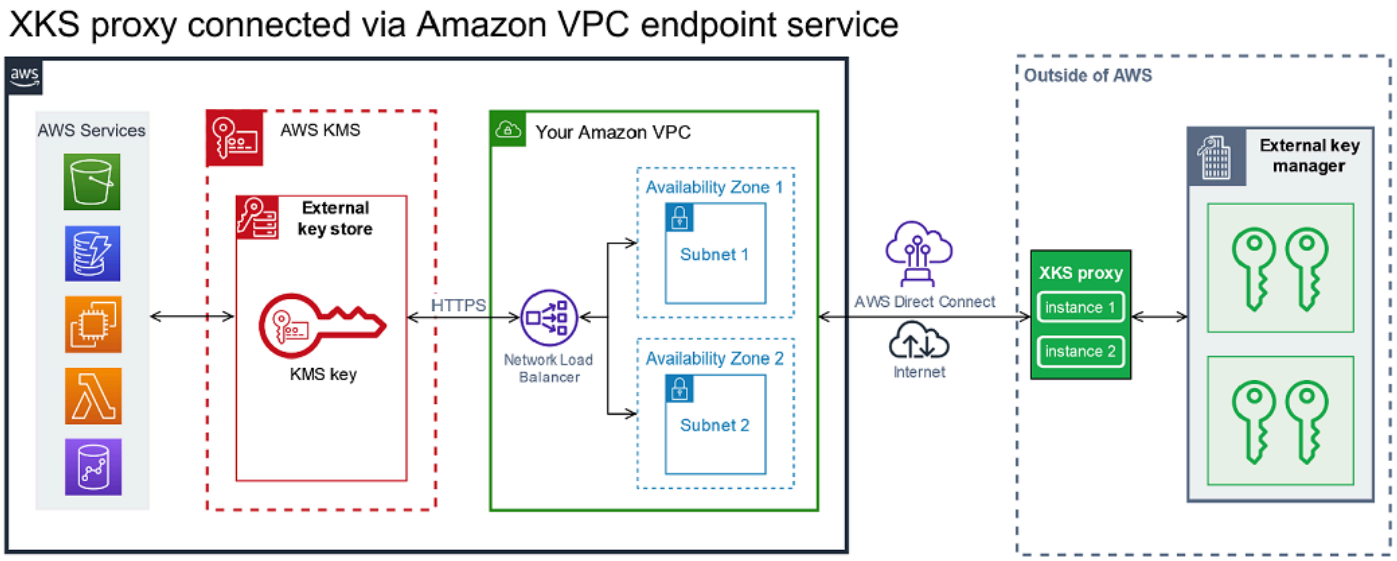
Votre service de point de terminaison peut utiliser n'importe laquelle des options Network-to-Amazon VPC (du réseau vers Amazon VPC) prises en charge pour les communications, notamment AWS Direct Connect.

Cette option de connectivité est plus compliquée à configurer et à gérer. Mais elle utilise AWS PrivateLink, ce qui permet à AWS KMS de se connecter en privé à votre Amazon VPC et à votre proxy de magasin de clés externe sans utiliser l'Internet public.

Vous pouvez localiser le proxy de votre magasin de clés externe dans votre Amazon VPC.



Vous pouvez également localiser le proxy de votre magasin de clés externe à l'extérieur d'AWS et n'utiliser votre service de point de terminaison d'un Amazon VPC que pour une communication sécurisée avec AWS KMS.



Configurer la connectivité au service de point de terminaison d'un VPC

Suivez les instructions de cette section pour créer et configurer les ressources AWS et les composants associés requis pour un magasin de clés externe utilisant la connectivité au service de point de terminaison d'un VPC. Les ressources répertoriées pour cette option de connectivité complètent les ressources requises pour tous les magasins de clés externes. Après avoir créé et configuré les ressources requises, vous pouvez créer votre magasin de clés externe.

Vous pouvez localiser votre proxy de magasin de clés externe dans votre Amazon VPC ou localiser le proxy en dehors d'AWS et utiliser votre service de point de terminaison d'un VPC pour la communication.

Avant de commencer, vérifiez que vous avez besoin d'un magasin de clés externe. La plupart des clients peuvent utiliser des clés KMS soutenues par des éléments de clé AWS KMS.

> **Note**
>
> Certains des éléments requis pour la connectivité au service de point de terminaison d'un VPC peuvent être inclus dans votre gestionnaire de clés externe. En outre, votre logiciel peut avoir des exigences de configuration supplémentaires. Avant de créer et de configurer les ressources AWS de cette section, consultez la documentation de votre proxy et de votre gestionnaire de clés.

Rubriques

- Exigences pour la connectivité au service de point de terminaison d'un VPC
- Créer un Amazon VPC et des sous-réseaux
- Créer un groupe cible
- Créer un équilibreur de charge réseau
- Créer un service de point de terminaison d'un VPC
- Vérifier votre nom de domaine DNS privé
- Autoriser AWS KMS à se connecter au service de point de terminaison d'un VPC

Exigences pour la connectivité au service de point de terminaison d'un VPC

Si vous choisissez la connectivité au service de point de terminaison d'un VPC pour votre magasin de clés externe, les ressources suivantes sont requises.

Pour minimiser la latence du réseau, créez vos composants AWS dans l'Région AWS prise en charge la plus proche de votre gestionnaire de clés externe. Si possible, choisissez une région dont le temps d'aller-retour sur le réseau (RTT, round-trip time) est inférieur ou égal à 35 millisecondes.

- Un Amazon VPC connecté à votre gestionnaire de clés externe. Il doit avoir au moins deux sous-réseaux privés dans deux zones de disponibilité différentes.

  Vous pouvez utiliser un Amazon VPC existant pour votre magasin de clés externe, à condition qu'il réponde aux exigences d'utilisation avec un magasin de clés externe. Plusieurs magasins de clés externes peuvent partager un Amazon VPC, mais chaque magasin de clés externe doit disposer de son propre service de point de terminaison d'un VPC et d'un nom DNS privé.
- Un service de point de terminaison d'un Amazon VPC à technologie AWS PrivateLink avec un équilibreur de charge réseau et un groupe cible.

  Le service de point de terminaison ne peut pas exiger d'acceptation. Vous devez également ajouter AWS KMS en tant que principal autorisé. Cela permet à AWS KMS de créer des points de terminaison d'interface afin qu'il puisse communiquer avec le proxy de votre magasin de clés externe.
- Un nom DNS privé pour le service de point de terminaison d'un VPC qui est unique dans sa Région AWS.

Le nom DNS privé doit être un sous-domaine d'un domaine public de niveau supérieur. Par exemple, si le nom DNS privé est `myproxy-private.xks.example.com`, il doit être un sous-domaine d'un domaine public tel que `xks.example.com` ou `example.com`.

Vous devez vérifier la propriété du domaine DNS pour le nom DNS privé.

- Un certificat TLS émis par une autorité de certification publique prise en charge pour votre proxy de magasin de clés externe.

  Le nom commun (CN) du sujet sur le certificat TLS doit correspondre au nom DNS privé. Par exemple, si le nom DNS privé est `myproxy-private.xks.example.com`, le CN du certificat TLS doit être `myproxy-private.xks.example.com` ou `*.xks.example.com`.

Pour connaître toutes les exigences relatives à un magasin de clés externe, veuillez consulter la rubrique Réunir les conditions préalables.

Créer un Amazon VPC et des sous-réseaux

La connectivité au service de point de terminaison d'un VPC nécessite un Amazon VPC connecté à votre gestionnaire de clés externe avec au moins deux sous-réseaux privés. Vous pouvez créer un Amazon VPC ou utiliser un Amazon VPC existant qui répond aux exigences relatives aux magasins de clés externes. Pour de plus amples informations sur la création d'un VPC, veuillez consulter la rubrique Créer un VPC dans le Guide de l'utilisateur Amazon Virtual Private Cloud.

Exigences pour votre Amazon VPC

Pour fonctionner avec des magasins de clés externes à l'aide de la connectivité au service de point de terminaison d'un VPC, l'Amazon VPC doit posséder les propriétés suivantes :

- Se trouver dans les mêmes Compte AWS et région prise en charge que votre magasin de clés externe.
- Comporter au moins deux sous-réseaux privés, chacun dans une zone de disponibilité différente.
- La plage d'adresses IP privées de votre Amazon VPC ne doit pas chevaucher la plage d'adresses IP privées du centre de données hébergeant votre gestionnaire de clés externe.
- Tous les composants doivent utiliser IPv4.

Vous disposez de nombreuses options pour connecter l'Amazon VPC à votre proxy de magasin de clés externe. Choisissez une option qui répond à vos exigences de performance et de sécurité.

Pour obtenir la liste, veuillez consulter la rubrique Connexion de votre VPC à des réseaux distants et Options de connectivité entre le réseau et Amazon VPC (langue française non garantie). Pour de plus amples informations, veuillez consulter AWS Direct Connect et le Guide de l'utilisateur AWS Site-to-Site VPN.

Créer un Amazon VPC pour votre magasin de clés externe

Utilisez les instructions suivantes pour créer l'Amazon VPC pour votre magasin de clés externe. Un Amazon VPC n'est requis que si vous choisissez l'option de connectivité au service de point de terminaison d'un VPC. Vous pouvez utiliser un Amazon VPC existant qui répond aux exigences d'un magasin de clés externe.

Suivez les instructions de la section Créer un VPC, des sous-réseaux et d'autres ressources VPC à l'aide des valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Bloc d'adresse CIDR IPv4	Saisissez les adresses IP de votre VPC. La plage d'adresses IP privées de votre Amazon VPC ne doit pas chevaucher la plage d'adresses IP privées du centre de données hébergeant votre gestionnaire de clés externe.
Nombre de zones de disponibilité (AZ)	2 ou plus
Nombre de sous-réseaux publics	Pas d'exigence minimale (0)
Nombre de sous-réseaux privés	Un pour chaque AZ
Passerelles NAT	Pas d'exigence minimale.

Champ	Valeur
Points de terminaison d'un VPC	Pas d'exigence minimale.
Enable DNS hostnames	Oui
Activer la résolution DNS	Oui

Assurez-vous de tester la communication de votre VPC. Par exemple, si le proxy de votre magasin de clés externe ne se trouve pas dans votre Amazon VPC, créez une instance Amazon EC2 dans votre Amazon VPC et vérifiez que l'Amazon VPC peut communiquer avec le proxy de votre magasin de clés externe.

Connecter le VPC au gestionnaire de clés externe

Connectez le VPC au centre de données qui héberge votre gestionnaire de clés externe en utilisant l'une des options de connectivité réseau prises en charge par Amazon VPC. Assurez-vous que l'instance Amazon EC2 du VPC (ou le proxy de magasin de clés externe, s'il se trouve dans le VPC) peut communiquer avec le centre de données et le gestionnaire de clés externe.

Créer un groupe cible

Avant de créer le service de point de terminaison d'un VPC requis, créez ses composants requis, un équilibreur de charge réseau (NLB) et un groupe cible. L'équilibreur de charge réseau (NLB) distribue les requêtes entre plusieurs cibles saines, chacune pouvant répondre à la requête. Au cours de cette étape, vous créez un groupe cible avec au moins deux hôtes pour votre proxy de magasin de clés externe et vous enregistrez vos adresses IP auprès du groupe cible.

Suivez les instructions de la section Configure a target group (Configurer un groupe cible) à l'aide des valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Type de cible	Adresses IP

Champ	Valeur
Protocole	TCP
Port	443
Type d'adresse IP	IPv4
VPC	Choisissez le VPC dans lequel vous allez créer le service de point de terminaison d'un VPC pour votre magasin de clés externe.
Protocole et chemin de surveillance de l'état	Votre protocole et votre chemin de surveillance de l'état varient en fonction de la configuration du proxy de votre magasin de clés externe. Consultez la documentation de votre gestionnaire de clés externe ou de votre proxy de magasin de clés externe. Pour des informations générales sur la configuration de la surveillance de l'état de vos groupes cibles, veuillez consulter la rubrique Health checks for your target groups (Surveillance de l'état de vos groupes cibles) dans le Guide de l'utilisateur Elastic Load Balancing pour les Network Load Balancers.
Réseau	Autre adresse IP privée
IPv4 address (Adresse IPv4)	Les adresses privées de votre proxy de magasin de clés externe
Ports	443

Créer un équilibreur de charge réseau

L'équilibreur de charge réseau distribue le trafic réseau, y compris les requêtes d'AWS KMS auprès de votre proxy de magasin de clés externe, aux cibles configurées.

Suivez les instructions de la rubrique Configure a load balancer and a listener (Configurer un équilibreur de charge et un écouteur) pour configurer et ajouter un écouteur et créer un équilibreur de charge en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Scheme	Internal (Interne)
Type d'adresse IP	IPv4
Mappage du réseau	Choisissez le VPC dans lequel vous allez créer le service de point de terminaison d'un VPC pour votre magasin de clés externe.
Mappage	Choisissez les deux zones de disponibilité (au moins deux) que vous avez configurées pour vos sous-réseaux VPC. Vérifiez les noms de sous-réseaux et l'adresse IP privée.
Protocole	TCP
Port	443
Action par défaut : Réacheminer vers	Choisissez le groupe cible pour votre équilibreur de charge réseau.

Créer un service de point de terminaison d'un VPC

En général, vous créez un point de terminaison vers un service. Toutefois, lorsque vous créez un service de point de terminaison d'un VPC, vous êtes le fournisseur et AWS KMS crée un point de terminaison vers votre service. Pour un magasin de clés externe, créez un service de point de terminaison d'un VPC à l'aide de l'équilibreur de charge réseau que vous avez créé à l'étape précédente. Le service de point de terminaison d'un VPC doit se trouver dans les mêmes Compte AWS et région prise en charge que votre magasin de clés externe.

Plusieurs magasins de clés externes peuvent partager un Amazon VPC, mais chaque magasin de clés externe doit disposer de son propre service de point de terminaison d'un VPC et d'un nom DNS privé.

Suivez les instructions de la rubrique Create an endpoint service (Créer un service de point de terminaison) pour créer votre service de point de terminaison d'un VPC avec les valeurs obligatoires

suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Nouveau type d'équilibreur de charge	Réseau
Équilibreurs de charge disponibles	Choisissez l'équilibreur de charge réseau que vous avez créé à l'étape précédente. Si votre nouvel équilibreur de charge ne figure pas dans la liste, vérifiez que son état est actif. Il peut s'écouler quelques minutes avant que l'état de l'équilibreur de charge ne passe d'alloué à actif.
Acceptation requise	Faux. Désactivez la case à cocher. N'exigez pas d'acceptation. AWS KMS ne peut pas se connecter au service de point de terminaison d'un VPC sans une acceptation manuelle. Si l'acceptation est requise, les tentatives de création du magasin de clés externe échouent avec une exception `XksProxyInvalidConfigurationException` .
Activer un nom DNS privé	Associez un nom DNS privé au service
Nom DNS privé	Saisissez un nom DNS privé unique dans sa Région AWS. Le nom DNS privé doit être un sous-domaine d'un domaine public de niveau supérieur. Par exemple, si le nom DNS privé est `myproxy-private.xks.example.com` , il doit être un sous-domaine d'un domaine public tel que `xks.example.com` ou `example.com` . Ce nom DNS privé doit correspondre au nom commun (CN) du sujet figurant dans le certificat TLS configuré sur le proxy de votre magasin de clés externe. Par exemple, si le nom DNS privé est `myproxy-private.xks.example.com` , le CN du certificat TLS doit être `myproxy-private.xks.example.com` ou `*.xks.example.com` .

Champ	Valeur
	Si le certificat et le nom DNS privé ne correspondent pas, les tentatives de connexion d'un magasin de clés externe à son proxy de magasin de clés externe échouent avec le code d'erreur de connexion de `XKS_PROXY_INVALID_TLS_CONFIGURATION`. Pour plus de détails, consultez Erreurs de configuration générale.
Types d'adresses IP pris en charge	IPv4

Vérifier votre nom de domaine DNS privé

Lorsque vous créez votre service de point de terminaison d'un VPC, son statut de vérification de domaine est `pendingVerification`. Avant d'utiliser le service de point de terminaison d'un VPC pour créer un magasin de clés externe, ce statut doit être `verified`. Pour vérifier que vous êtes bien le propriétaire du domaine associé à votre nom DNS privé, vous devez créer un enregistrement TXT sur un serveur DNS public.

Par exemple, si le nom DNS privé de votre service de point de terminaison d'un VPC est `myproxy-private.xks.example.com`, vous devez créer un enregistrement TXT dans un domaine public, tel que `xks.example.com` ou `example.com`, selon ce qui est public. AWS PrivateLink recherche d'abord l'enregistrement TXT sur `xks.example.com` puis sur `example.com`.

> **Tip**
>
> Après avoir ajouté un enregistrement TXT, il peut s'écouler quelques minutes avant que la valeur du Domain verification status (Statut de vérification du domaine) passe de `pendingVerification` à `verify`.

Pour commencer, identifiez le statut de vérification de votre domaine à l'aide de l'une des méthodes suivantes. Les valeurs valides sont `verified`, `pendingVerification` et `failed`.

- Dans la console Amazon VPC, choisissez Endpoint services (Services de points de terminaison), puis choisissez votre service de point de terminaison. Dans le volet d'informations, veuillez consulter la rubrique Domain verification status (Statut de vérification du domaine).

- Utilisez l'DescribeVpcEndpointServiceConfigurationsopération. La valeur de `State` se trouve dans le champ `ServiceConfigurations.PrivateDnsNameConfiguration.State`.

Si le statut de vérification n'est pas `verified`, suivez les instructions de la rubrique Domain ownership verification (Vérification de la propriété du domaine) pour ajouter un enregistrement TXT au serveur DNS de votre domaine et vérifier que l'enregistrement TXT est publié. Vérifiez ensuite à nouveau votre statut de vérification.

Vous n'êtes pas obligé de créer un enregistrement A pour le nom de domaine DNS privé. Lorsqu'AWS KMS crée un point de terminaison d'interface vers votre service de point de terminaison d'un VPC, AWS PrivateLink crée automatiquement une zone hébergée avec l'enregistrement A requis pour le nom de domaine privé dans le VPC AWS KMS. Pour les magasins de clés externes dotés d'une connectivité au service de point de terminaison d'un VPC, cela se produit lorsque vous connectez votre magasin de clés externe à son proxy de magasin de clés externe.

Autoriser AWS KMS à se connecter au service de point de terminaison d'un VPC

Vous devez ajouter AWS KMS à la liste Allow principals (Principaux autorisés) pour votre service de point de terminaison d'un VPC. Cela permet à AWS KMS de créer des points de terminaison d'interface vers votre service de point de terminaison d'un VPC. Si AWS KMS n'est pas un principal autorisé, les tentatives de création d'un magasin de clés externe échoueront avec une exception `XksProxyVpcEndpointServiceNotFoundException`.

Suivez les instructions de la rubrique Manage permissions (Gérer les autorisations) du Guide AWS PrivateLink. Utilisez la valeur obligatoire suivante.

Champ	Valeur
ARN	`cks.kms.<region>.amazonaws.com` Par exemple, `cks.kms.us-east-1.amazonaws.com`

Suivant : Créer un magasin de clés externe

Gérer un magasin de clés externe

Vous pouvez gérer un magasin de clés externe à l'aide de la console AWS KMS ou de l'API AWS KMS. Vous pouvez créer un magasin de clés externe, afficher et modifier ses propriétés, surveiller

ses performances, le connecter et le déconnecter de son proxy de magasin de clés externe et supprimer le magasin de clés externe.

Rubriques

- Créer un magasin de clés externe
- Modifier les propriétés du magasin de clés externe
- Afficher un magasin de clés externe
- Surveiller un magasin de clés externe
- Connecter et déconnecter un magasin de clés externe
- Supprimer un magasin de clés externe

Créer un magasin de clés externe

Vous pouvez créer un ou plusieurs magasins de clés externes dans chaque Compte AWS et région. Chaque magasin de clés externe doit être associé à un gestionnaire de clés externe en dehors d'AWS et à un proxy de magasin de clés externe (proxy XKS) qui assure la médiation des communications entre AWS KMS et votre gestionnaire de clés externe. Pour plus de détails, consultez Planifier un magasin de clés externe. Avant de commencer, vérifiez que vous avez besoin d'un magasin de clés externe. La plupart des clients peuvent utiliser des clés KMS soutenues par des éléments de clé AWS KMS.

> **Tip**
>
> Certains gestionnaires de clés externes proposent une méthode plus simple pour créer un magasin de clés externe. Pour en savoir plus, veuillez consulter la documentation de votre gestionnaire de clés externe.

Avant de créer votre magasin de clés externe, vous devez réunir les conditions préalables. Au cours du processus de création, vous définissez les propriétés de votre magasin de clés externe. Plus important encore, vous indiquez si votre magasin de clés externe dans AWS KMS utilise un point de terminaison public ou un service de point de terminaison d'un VPC pour se connecter à son proxy de magasin de clés externe. Vous spécifiez également les détails de la connexion, y compris le point de terminaison de l'URI du proxy et le chemin au sein de ce point de terminaison du proxy où AWS KMS envoie les requêtes d'API au proxy.

- Si vous utilisez une connectivité au point de terminaison public, assurez-vous qu'AWS KMS peut communiquer avec votre proxy via Internet à l'aide d'une connexion HTTPS. Cela implique de configurer le protocole TLS sur le proxy de magasin de clés externe et de s'assurer que tous les pare-feux situés entre AWS KMS et le proxy autorisent le trafic en provenance et à destination du port 443 sur le proxy. Lors de la création d'un magasin de clés externe avec une connectivité au point de terminaison public, AWS KMS teste la connexion en envoyant une requête d'état au proxy de magasin de clés externe. Ce test vérifie que le point de terminaison est accessible et que votre proxy de magasin de clés externe acceptera une requête signée avec vos informations d'identification pour l'authentification du proxy de magasin de clés externe. Si cette requête de test échoue, l'opération de création du magasin de clés externe échoue.
- Si vous utilisez la connectivité au service de point de terminaison d'un VPC, assurez-vous que l'équilibreur de charge réseau, le nom DNS privé et le service de point de terminaison d'un VPC sont correctement configurés et opérationnels. Si le proxy de magasin de clés externe ne se trouve pas dans le VPC, vous devez vous assurer que le service de point de terminaison d'un VPC peut communiquer avec le proxy de magasin de clés externe. (AWS KMS teste la connectivité au service de point de terminaison d'un VPC lorsque vous connectez le magasin de clés externe à son proxy de magasin de clés externe.)

Considérations supplémentaires :

- AWS KMSenregistre les CloudWatch statistiques et les dimensions d'Amazon, en particulier pour les principaux magasins externes. Des graphiques de surveillance basés sur certaines de ces métriques apparaissent dans la console AWS KMS pour chaque magasin de clés externe. Nous vous recommandons vivement d'utiliser ces métriques afin de créer des alertes qui surveillent votre magasin de clés externe. Ces alertes vous préviennent des signes précoces de problèmes de performance et de fonctionnement avant qu'ils ne se produisent. Pour obtenir des instructions, veuillez consulter Surveiller un magasin de clés externe.
- Les magasins de clés externes sont soumis à des quotas de ressources. L'utilisation de clés KMS dans un magasin de clés externe est soumise à des quotas de requêtes. Passez en revue ces quotas avant de concevoir l'implémentation de votre magasin de clés externe.

> **Note**
>
> Vérifiez votre configuration pour détecter les dépendances circulaires susceptibles de l'empêcher de fonctionner.

Par exemple, si vous créez votre proxy de stockage de clés externe à l'aide de AWS ressources, assurez-vous que le fonctionnement du proxy ne nécessite pas la disponibilité d'une clé KMS dans un magasin de clés externe accessible via ce proxy.

Tous les nouveaux magasins de clés externes sont créés dans un état déconnecté. Avant de créer des clés KMS dans votre magasin de clés externe, vous devez le connecter à son proxy de magasin de clés externe. Pour modifier les propriétés de votre magasin de clés externe, modifiez les paramètres de votre magasin de clés externe.

Rubriques

- Rassembler les conditions requises
- Fichier de configuration du proxy
- Créer un magasin de clés externe (console)
- Créer un magasin de clés externe (API)

Rassembler les conditions requises

Avant de créer un magasin de clés externe, vous devez réunir les composants requis, notamment le gestionnaire de clés externe que vous utiliserez pour prendre en charge le magasin de clés externe et le proxy de magasin de clés externe qui traduit les requêtes AWS KMS dans un format compréhensible par votre gestionnaire de clés externe.

Les composants suivants sont requis pour tous les magasins de clés externes. Outre ces composants, vous devez fournir les composants qui prennent en charge l'option de connectivité par proxy de magasin de clés externe que vous choisissez.

Tip

Votre gestionnaire de clés externe peut inclure certains de ces composants, ou ils peuvent être configurés pour vous. Pour en savoir plus, veuillez consulter la documentation de votre gestionnaire de clés externe.
Si vous créez votre magasin de clés externe dans la console AWS KMS, vous pouvez charger un fichier de configuration de proxy JSON qui spécifie le chemin d'URI de proxy et les informations d'identification pour l'authentification du proxy. Certains proxys de magasin de clés externe génèrent ce fichier pour vous. Pour plus de détails, veuillez consulter la

documentation de votre proxy de magasin de clés externe ou de votre gestionnaire de clés externe.

Gestionnaire de clés externe

Chaque magasin de clés externe nécessite au moins une instance de gestionnaire de clés externe. Il peut s'agir d'un module de sécurité matérielle (HSM) physique ou virtuel ou d'un logiciel de gestion des clés.

Vous pouvez utiliser un seul gestionnaire de clés, mais nous recommandons au moins deux instances de gestionnaire de clés connexes qui partagent des clés cryptographiques pour des raisons de redondance. Le magasin de clés externe ne nécessite pas l'utilisation exclusive du gestionnaire de clés externe. Toutefois, le gestionnaire de clés externe doit être capable de gérer la fréquence attendue des requêtes de chiffrement et de déchiffrement émanant des services AWS qui utilisent des clés KMS dans le magasin de clés externe pour protéger vos ressources. Votre gestionnaire de clés externe doit être configuré pour traiter jusqu'à 1 800 requêtes par seconde et pour répondre dans le délai d'expiration de 250 millisecondes pour chaque requête. Nous vous recommandons de placer le gestionnaire de clés externe à proximité d'une Région AWS de manière à ce que le temps d'aller-retour sur le réseau (RTT) soit inférieur ou égal à 35 millisecondes.

Si le proxy de votre magasin de clés externe le permet, vous pouvez modifier le gestionnaire de clés externe que vous associez à votre proxy de magasin de clés externe, mais le nouveau gestionnaire de clés externe doit être une sauvegarde ou un instantané contenant les mêmes éléments de clé. Si la clé externe que vous associez à une clé KMS n'est plus disponible pour le proxy de votre magasin de clés externe, AWS KMS ne peut pas déchiffrer le texte chiffré qui a été chiffré avec la clé KMS.

Le gestionnaire de clés externe doit être accessible au proxy de magasin de clés externe. Si la GetHealthStatusréponse du proxy indique que toutes les instances du gestionnaire de clés externe le sontUnavailable, toutes les tentatives de création d'une banque de clés externe échouent avec un XksProxyUriUnreachableException.

Proxy de magasin de clés externe

Vous devez spécifier un proxy de magasin de clés externe (proxy XKS) conforme aux exigences de conception de spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie). Vous pouvez développer ou acheter un proxy de magasin de clés externe, ou utiliser un proxy de magasin de clés externe fourni par ou intégré à votre gestionnaire de clés externe. AWS KMS recommande que votre proxy de magasin de clés externe soit configuré pour

traiter jusqu'à 1 800 requêtes par seconde et répondre dans le délai d'expiration de 250 millisecondes pour chaque requête. Nous vous recommandons de placer le gestionnaire de clés externe à proximité d'une Région AWS de manière à ce que le temps d'aller-retour sur le réseau (RTT) soit inférieur ou égal à 35 millisecondes.

Vous pouvez utiliser un proxy de magasin de clés externe pour plusieurs magasins de clés externes, mais chaque magasin de clés externe doit disposer d'un point de terminaison et d'un chemin d'URI uniques au sein du proxy de magasin de clés externe pour ses requêtes.

Si vous utilisez la connectivité au service de point de terminaison d'un VPC, vous pouvez localiser le proxy de votre magasin de clés externe dans votre Amazon VPC, mais cela n'est pas obligatoire. Vous pouvez localiser votre proxy à l'extérieur d'AWS, par exemple dans votre centre de données privé, et utiliser le service de point de terminaison d'un VPC uniquement pour communiquer avec le proxy.

Informations d'identification pour l'authentification du proxy

Pour créer un magasin de clés externe, vous devez spécifier vos informations d'identification pour l'authentification du proxy de magasin de clés externe (`XksProxyAuthenticationCredential`).

Vous devez définir des informations d'identification pour l'authentification (`XksProxyAuthenticationCredential`) pour AWS KMS sur votre proxy de magasin de clés externe. AWS KMS s'authentifie auprès de votre proxy en signant ses requêtes à l'aide du processus Signature version 4 (SigV4) en utilisant les informations d'identification pour l'authentification du proxy de magasin de clés externe. Vous spécifiez les informations d'identification pour l'authentification lorsque vous créez votre magasin de clés externe et vous pouvez les modifier à tout moment. Si votre proxy effectue une rotation de vos informations d'identification, veillez à mettre à jour les valeurs d'informations d'identification de votre magasin de clés externe.

Les informations d'identification pour l'authentification du proxy comportent deux parties. Vous devez fournir les deux parties pour votre magasin de clés externe.

- ID de la clé d'accès : identifie la clé d'accès secrète. Vous pouvez fournir cet ID en texte brut.
- Clé d'accès secrète : partie secrète des informations d'identification. AWS KMS chiffre la clé d'accès secrète contenue dans les informations d'identification avant de les stocker.

Les informations d'identification SigV4 qu'AWS KMS utilise pour signer les requêtes adressées au proxy de magasin de clés externe ne sont pas liées aux informations d'identification SigV4 associées

aux principaux AWS Identity and Access Management de vos comptes AWS. Ne réutilisez aucune information d'identification IAM SigV4 pour votre proxy de magasin de clés externe.

Connectivité de proxy

Pour créer un magasin de clés externe, vous devez spécifier l'option de connectivité de proxy de votre magasin de clés externe (`XksProxyConnectivity`).

AWS KMS peut communiquer avec le proxy de votre magasin de clés externe en utilisant un point de terminaison public ou un service de point de terminaison Amazon Virtual Private Cloud (Amazon VPC). Bien qu'un point de terminaison public soit plus simple à configurer et à gérer, il se peut qu'il ne réponde pas aux exigences de sécurité de chaque installation. Si vous choisissez l'option de connectivité au service de point de terminaison d'un Amazon VPC, vous devez créer et gérer les composants requis, notamment un Amazon VPC avec au moins deux sous-réseaux dans deux zones de disponibilité différentes, un service de point de terminaison d'un VPC avec un équilibreur de charge réseau et un groupe cible, ainsi qu'un nom DNS privé pour le service de point de terminaison d'un VPC.

Vous pouvez modifier l'option de connectivité de proxy pour votre magasin de clés externe. Toutefois, vous devez vous assurer de la disponibilité continue des éléments de clé associés aux clés KMS dans votre magasin de clés externe. Sinon, AWS KMS ne peut pas déchiffrer le texte chiffré avec ces clés KMS.

Pour savoir quelle option de connectivité de proxy convient le mieux à votre magasin de clés externe, veuillez consulter la rubrique Choisir une option de connectivité de proxy. Pour obtenir de l'aide sur la création et la configuration de la connectivité au service de point de terminaison d'un VPC, veuillez consulter la rubrique Configurer la connectivité au service de point de terminaison d'un VPC.

Point de terminaison d'URI de proxy

Pour créer un magasin de clés externe, vous devez spécifier le point de terminaison (`XksProxyUriEndpoint`) qu'AWS KMS utilise pour envoyer des requêtes au proxy de magasin de clés externe.

Le protocole doit être HTTPS. AWS KMS communique sur le port 443. Ne spécifiez pas le port dans la valeur de point de terminaison d'URI de proxy.

- Connectivité des points de terminaison publics : spécifiez le point de terminaison accessible au public pour votre proxy de magasin de clés externe. Ce point de terminaison doit être accessible avant de créer votre magasin de clés externe.

- Connectivité au service de point de terminaison d'un VPC : spécifiez `https://` suivi du nom DNS privé du service de point de terminaison d'un VPC.

Le certificat de serveur TLS configuré sur le proxy de magasin de clés externe doit correspondre au nom de domaine indiqué dans le point de terminaison de l'URI de proxy de magasin de clés externe et être émis par une autorité de certification prise en charge pour les magasins de clés externes. Pour obtenir une liste, veuillez consulter les Autorités de certification approuvées (langue française non garantie). Votre autorité de certification exigera une preuve de propriété du domaine avant de délivrer le certificat TLS.

Le nom commun (CN) du sujet sur le certificat TLS doit correspondre au nom DNS privé. Par exemple, si le nom DNS privé est `myproxy-private.xks.example.com`, le CN du certificat TLS doit être `myproxy-private.xks.example.com` ou `*.xks.example.com`.

Vous pouvez modifier le point de terminaison de l'URI de votre proxy, mais assurez-vous que le proxy de magasin de clés externe a accès aux éléments de clé associés aux clés KMS de votre magasin de clés externe. Sinon, AWS KMS ne peut pas déchiffrer le texte chiffré avec ces clés KMS.

Exigences relatives à l'unicité

- La combinaison du point de terminaison d'URI de proxy (`XksProxyUriEndpoint`) et de la valeur du chemin d'URI de proxy (`XksProxyUriPath`) doit être unique dans l'Compte AWS et la région.
- Les magasins de clés externes connectés à un point de terminaison public peuvent partager le même point de terminaison d'URI de proxy, à condition qu'ils aient des valeurs de chemin d'URI de proxy différentes.
- Un magasin de clés externe connecté à un point de terminaison public ne peut pas utiliser une valeur de point de terminaison d'URI de proxy identique à celle d'un magasin de clés externe connecté aux services de point de terminaison d'un VPC dans la même Région AWS, même si les magasins de clés se trouvent dans des Comptes AWS différents.
- Chaque magasin de clés externe connecté à un point de terminaison d'un VPC doit avoir son propre nom DNS privé. Le point de terminaison d'URI de proxy (nom DNS privé) doit être unique dans l'Compte AWS et la région.

Chemin d'URI de proxy

Pour créer un magasin de clés externe, vous devez spécifier le chemin de base vers les API de proxy requises dans votre proxy de magasin de clés externe. La valeur doit commencer par `/` et se terminer

par /kms/xks/v1, où `v1` représente la version de l'API AWS KMS pour le proxy de magasin de clés externe. Ce chemin peut inclure un préfixe facultatif entre les éléments requis tels que `/example-prefix/kms/xks/v1`. Pour trouver cette valeur, veuillez consulter la documentation de votre proxy de magasin de clés externe.

AWS KMS envoie des requêtes de proxy à l'adresse spécifiée par la concaténation du point de terminaison d'URI de proxy et du chemin d'URI de proxy. Par exemple, si le point de terminaison d'URI de proxy est `https://myproxy.xks.example.com` et que le chemin d'URI de proxy est `/kms/xks/v1`, AWS KMS envoie ses requêtes d'API proxy à `https://myproxy.xks.example.com/kms/xks/v1`.

Vous pouvez modifier le chemin d'URI de votre proxy, mais assurez-vous que le proxy de magasin de clés externe a accès aux éléments de clé associés aux clés KMS de votre magasin de clés externe. Sinon, AWS KMS ne peut pas déchiffrer le texte chiffré avec ces clés KMS.

Exigences relatives à l'unicité

- La combinaison du point de terminaison d'URI de proxy (`XksProxyUriEndpoint`) et de la valeur du chemin d'URI de proxy (`XksProxyUriPath`) doit être unique dans l'Compte AWS et la région.

Service de point de terminaison d’un VPC

Spécifie le nom du service de point de terminaison d'un Amazon VPC utilisé pour communiquer avec le proxy de votre magasin de clés externe. Ce composant n'est requis que pour les magasins de clés externes qui utilisent la connectivité au service de point de terminaison d'un VPC. Pour obtenir de l'aide sur la configuration de votre service de point de terminaison d'un VPC pour un magasin de clés externe, veuillez consulter la rubrique Configurer la connectivité au service de point de terminaison d'un VPC.

Le service de point de terminaison d'un VPC doit posséder les propriétés suivantes :

- Le service de point de terminaison d'un VPC doit se trouver dans les mêmes Compte AWS et région que le magasin de clés externe.
- Il doit comporter un équilibreur de charge réseau (NLB) connecté à au moins deux sous-réseaux, dans deux zones de disponibilités distinctes.
- La liste des principaux autorisés pour le service de point de terminaison d'un VPC doit inclure le principal de service AWS KMS pour la région : `cks.kms.<region>.amazonaws.com`, tel que `cks.kms.us-east-1.amazonaws.com`.

- L'acceptation des requêtes de connexion ne doit pas être requise.
- Il doit avoir un nom DNS privé dans un domaine public de niveau supérieur. Par exemple, vous pouvez avoir un nom DNS privé myproxy-private.xks.example.com dans le domaine public `xks.example.com`.

  Le nom DNS privé d'un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC doit être unique dans sa Région AWS.
- L'état de vérification du domaine du nom DNS privé doit être `verified`.
- Le certificat de serveur TLS configuré sur le proxy de magasin de clés externe doit spécifier le nom d'hôte DNS privé auquel le point de terminaison est accessible.

Exigences relatives à l'unicité

- Les magasins de clés externes connectés à des points de terminaison d'un VPC peuvent partager un `Amazon VPC`, mais chaque magasin de clés externe doit disposer de son propre service de point de terminaison d'un VPC et d'un nom DNS privé.

Fichier de configuration du proxy

Un fichier de configuration de proxy est un fichier JSON facultatif qui contient des valeurs pour le chemin d'URI de proxy et les propriétés d'informations d'identification pour l'authentification du proxy de votre magasin de clés externe. Lorsque vous créez ou modifiez un magasin de clés externe dans la console AWS KMS, vous pouvez télécharger un fichier de configuration de proxy pour fournir des valeurs de configuration pour votre magasin de clés externe. L'utilisation de ce fichier évite les erreurs de saisie et de collage et garantit que les valeurs de votre magasin de clés externe correspondent à celles de votre proxy de magasin de clés externe.

Les fichiers de configuration du proxy sont générés par le proxy de magasin de clés externe. Pour savoir si votre proxy de magasin de clés externe propose un fichier de configuration de proxy, veuillez consulter la documentation relative à votre proxy de magasin de clés externe.

Voici un exemple de fichier de configuration de proxy correctement formaté avec des valeurs fictives.

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9MO9+vE="
```

```
  }
}
```

Vous ne pouvez charger un fichier de configuration du proxy que lors de la création ou de la modification d'un magasin de clés externe dans la console AWS KMS. Vous ne pouvez pas l'utiliser avec les UpdateCustomKeyStoreopérations CreateCustomKeyStoreor, mais vous pouvez utiliser les valeurs du fichier de configuration du proxy pour vous assurer que les valeurs de vos paramètres sont correctes.

Créer un magasin de clés externe (console)

Avant de créer un magasin de clés externe, consultez Planifier un magasin de clés externe, choisissez votre type de connectivité de proxy et assurez-vous d'avoir créé et configuré tous les composants requis. Si vous avez besoin d'aide pour trouver l'une des valeurs requises, consultez la documentation de votre proxy de magasin de clés externe ou de votre logiciel de gestion des clés.

> **Note**
>
> Lorsque vous créez un magasin de clés externe dans l'AWS Management Console, vous pouvez télécharger un fichier de configuration de proxy JSON contenant des valeurs pour le chemin d'URI de proxy et les informations d'identification pour l'authentification du proxy. Certains proxys génèrent ce fichier pour vous. Il n'est pas obligatoire.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Choisissez Create external key store (Créer un magasin de clés externe).
5. Saisissez un nom convivial pour le magasin de clés externe. Le nom doit être unique parmi tous les magasins de clés externes de votre compte.

> **Important**
>
> N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

6. Choisissez votre type de connectivité de proxy.

   Votre choix de connectivité de proxy détermine les composants requis pour votre proxy de magasin de clés externe. Pour obtenir de l'aide pour faire ce choix, veuillez consulter la rubrique Choisir une option de connectivité de proxy.

7. Choisissez ou saisissez le nom du service de point de terminaison d'un VPC pour ce magasin de clés externe. Cette étape s'affiche uniquement lorsque le type de connectivité du proxy de votre magasin de clés externe est VPC endpoint service (Service de point de terminaison d'un VPC).

   Le service de point de terminaison d'un VPC et ses VPC doivent répondre aux exigences d'un magasin de clés externe. Pour plus de détails, consultez the section called “Rassembler les conditions requises”.

8. Saisissez votre point de terminaison d'URI de proxy. Le protocole doit être HTTPS. AWS KMS communique sur le port 443. Ne spécifiez pas le port dans la valeur de point de terminaison d'URI de proxy.

   Si AWS KMS reconnaît le service de point de terminaison d'un VPC que vous avez spécifié à l'étape précédente, il complète ce champ à votre place.

   Pour la connectivité au point de terminaison public, saisissez un URI de point de terminaison accessible au public. Pour la connectivité au point de terminaison d'un VPC, saisissez `https://` suivi du nom DNS privé du service de point de terminaison d'un VPC.

9. Pour saisir les valeurs du préfixe du chemin d'URI de proxy et des informations d'identification pour l'authentification du proxy, chargez un fichier de configuration de proxy ou saisissez les valeurs manuellement.

   - Si vous disposez d'un fichier de configuration de proxy facultatif contenant des valeurs pour le chemin d'URI de votre proxy et les informations d'identification pour l'authentification du proxy, choisissez Upload configuration file (Charger le fichier de configuration). Suivez les instructions pour charger le fichier.

Lorsque le fichier est chargé, la console affiche les valeurs du fichier dans des champs modifiables. Vous pouvez changer les valeurs maintenant ou modifier ces valeurs après la création du magasin de clés externe.

Pour afficher la valeur de la clé d'accès secrète, choisissez Show secret access key (Afficher la clé d'accès secrète).

- Si vous ne disposez pas d'un fichier de configuration du proxy, vous pouvez saisir manuellement le chemin d'URI de proxy et les valeurs d'informations d'identification pour l'authentification du proxy.

  a. Si vous n'avez pas de fichier de configuration de proxy, vous pouvez saisir manuellement l'URI de votre proxy. La console fournit la valeur /kms/xks/v1 requise.

     Si votre chemin d'URI de proxy inclut un préfixe facultatif, tel que l'`example-prefix` dans `/example-prefix/kms/xks/v1`, saisissez le préfixe dans le champ Proxy URI path prefix (Préfixe du chemin d'URI de proxy). Sinon, laissez le champ vide.

  b. Si vous ne disposez pas d'un fichier de configuration du proxy, vous pouvez saisir vos informations d'identification pour l'authentification du proxy manuellement. L'ID de clé d'accès et la clé d'accès secrète sont tous deux requis.

     - Dans Proxy credential: Access key ID (Informations d'identification du proxy : identifiant de la clé d'accès), saisissez l'ID de clé d'accès des informations d'identification pour l'authentification du proxy. L'ID de clé d'accès identifie la clé d'accès secrète.
     - Dans Proxy credential: Secret access key (Informations d'identification du proxy : clé d'accès secrète), saisissez la clé d'accès secrète des informations d'identification pour l'authentification du proxy.

     Pour afficher la valeur de la clé d'accès secrète, choisissez Show secret access key (Afficher la clé d'accès secrète).

     Cette procédure ne définit ni ne modifie les informations d'identification pour l'authentification que vous avez établies sur votre proxy de magasin de clés externe. Elle associe simplement ces valeurs à votre magasin de clés externe. Pour plus d'informations sur la définition, la modification et la rotation de vos informations d'identification pour l'authentification du proxy, veuillez consulter la documentation de votre proxy de magasin de clés externe ou de votre logiciel de gestion des clés.

Si vos informations d'identification pour l'authentification du proxy changent, modifiez le paramètre des informations d'identification de votre magasin de clés externe.

10. Choisissez Create external key store (Créer un magasin de clés externe).

Lorsque la procédure se termine avec succès, le nouveau magasin de clés externe s'affiche dans la liste des magasins de clés externes du compte et de la région. S'il ne réussit pas, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez CreateKey erreurs pour la clé externe.

Suivant : les nouveaux magasins de clés externes ne sont pas automatiquement connectés. Avant de créer les AWS KMS keys dans votre magasin de clés externe, vous devez connecter le magasin de clés externe à son proxy de magasin de clés externe.

Créer un magasin de clés externe (API)

Vous pouvez utiliser cette CreateCustomKeyStoreopération pour créer un nouveau magasin de clés externe. Pour obtenir de l'aide pour trouver les valeurs des paramètres requis, veuillez consulter la documentation de votre proxy de magasin de clés externe ou de votre logiciel de gestion des clés.

> **Tip**
>
> Vous ne pouvez pas charger de fichier de configuration de proxy lors de l'utilisation de l'opération `CreateCustomKeyStore`. Vous pouvez toutefois utiliser les valeurs du fichier de configuration de proxy pour vous assurer que les valeurs de vos paramètres sont correctes.

Pour créer un magasin de clés externe, l'opération `CreateCustomKeyStore` nécessite les valeurs de paramètres suivantes.

- `CustomKeyStoreName` : un nom convivial pour le magasin de clés externe qui est unique dans le compte.

  > **Important**
  >
  > N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.

- `CustomKeyStoreType` : spécifiez `EXTERNAL_KEY_STORE`.

- XksProxyConnectivity : spécifiez PUBLIC_ENDPOINT ou VPC_ENDPOINT_SERVICE.
- XksProxyAuthenticationCredential : spécifiez à la fois l'ID de clé d'accès et la clé d'accès secrète.
- XksProxyUriEndpoint : le point de terminaison qu'AWS KMS utilise pour communiquer avec votre proxy de magasin de clés externe.
- XksProxyUriPath : le chemin d'accès aux API de proxy au sein du proxy.
- XksProxyVpcEndpointServiceName : obligatoire uniquement lorsque la valeur de votre XksProxyConnectivity est VPC_ENDPOINT_SERVICE.

> **Note**
>
> Si vous utilisez l'AWS CLI version 1.0, exécutez la commande suivante avant de spécifier un paramètre avec une valeur HTTP ou HTTPS, tel que le paramètre XksProxyUriEndpoint.
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> Sinon, l'AWS CLI version 1.0 remplace la valeur du paramètre par le contenu trouvé à cette adresse URI, provoquant l'erreur suivante :
>
> ```
> Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
> https:// : received non 200 status code of 404
> ```

Les exemples suivants utilisent des valeurs fictives. Avant d'exécuter la commande, remplacez-les par des valeurs valides pour votre magasin de clés externe.

Créez un magasin de clés externe avec une connectivité au point de terminaison public.

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStorePublic \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity PUBLIC_ENDPOINT \
        --xks-proxy-uri-endpoint https://myproxy.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

Créez un magasin de clés externe avec une connectivité au service de point de terminaison d'un VPC.

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStoreVPC \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
        --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-svc-
example \
        --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

Lorsque l'opération est réussie, `CreateCustomKeyStore` renvoie l'ID du magasin de clés personnalisé, comme illustré dans l'exemple de réponse suivant.

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Si l'opération échoue, corrigez l'erreur indiquée par l'exception, puis réessayez. Pour obtenir de l'aide supplémentaire, consultez Résoudre les problèmes liés aux magasins de clés externes.

Suivant : pour utiliser le magasin de clés externe, connectez-le à son proxy de magasin de clés externe.

Modifier les propriétés du magasin de clés externe

Vous pouvez modifier les propriétés sélectionnées d'un magasin de clés externe existant.

Vous pouvez modifier certaines propriétés lorsque le magasin de clés externe est connecté ou déconnecté. Pour les autres propriétés, vous devez d'abord déconnecter votre magasin de clés externe de son proxy de magasin de clés externe. L'état de connexion du magasin de clés externe doit être `DISCONNECTED`. Lorsque votre magasin de clés externe est déconnecté, vous pouvez gérer le magasin de clés et ses clés KMS, mais vous ne pouvez pas créer ou utiliser des clés KMS dans le magasin de clés externe. Pour connaître l'état de connexion de votre magasin de clés externe, utilisez l'DescribeCustomKeyStoresopération ou consultez la section Configuration générale sur la page détaillée du magasin de clés externe.

Avant de mettre à jour les propriétés de votre magasin de clés externe, AWS KMS envoie une GetHealthStatusdemande au proxy du magasin de clés externe en utilisant les nouvelles valeurs. Si la requête aboutit, cela indique que vous pouvez vous connecter et vous authentifier à un proxy de magasin de clés externe avec les valeurs de propriété mises à jour. Si la requête échoue, l'opération de modification échoue avec une exception qui identifie l'erreur.

Lorsque l'opération de modification est terminée, les valeurs de propriété mises à jour pour votre magasin de clés externe apparaissent dans la console AWS KMS et dans la réponse de `DescribeCustomKeyStores`. Toutefois, il peut s'écouler jusqu'à cinq minutes avant que les modifications ne soient pleinement effectives.

Si vous modifiez votre magasin de clés externe dans la console AWS KMS, vous pouvez charger un fichier de configuration de proxy JSON qui spécifie le chemin d'URI de proxy et les informations d'identification pour l'authentification du proxy. Certains proxys de magasin de clés externe génèrent ce fichier pour vous. Pour plus de détails, veuillez consulter la documentation de votre proxy de magasin de clés externe ou de votre gestionnaire de clés externe.

> **Warning**
>
> Les valeurs de propriété mises à jour doivent connecter votre magasin de clés externe à un proxy pour le même gestionnaire de clés externe que celui utilisé dans les valeurs précédentes, ou pour une sauvegarde ou un instantané du gestionnaire de clés externe avec les mêmes clés cryptographiques. Si votre magasin de clés externe perd définitivement l'accès aux clés externes associées à ses clés KMS, le texte chiffré qui a été chiffré au moyen de ces clés externes est irrécupérable. En particulier, la modification de la connectivité de proxy d'un magasin de clés externe peut empêcher AWS KMS d'accéder à vos clés externes.

> **Tip**
>
> Certains gestionnaires de clés externes proposent une méthode plus simple pour modifier les propriétés du magasin de clés externe. Pour en savoir plus, veuillez consulter la documentation de votre gestionnaire de clés externe.

Vous pouvez modifier les propriétés suivantes d'un magasin de clés externe.

Propriétés du magasin de clés externe modifiables	Tout état de connexion	Exiger l'état Déconnecté
Nom du magasin de clés personnalisé Un nom convivial requis pour un magasin de clés personnalisé. ⚠ Important N'incluez pas d'informations confidentielles ou sensibles dans ce champ. Ce champ peut être affiché en texte brut dans les CloudTrail journaux et autres sorties.	✓	
Identifiant d'authentification du proxy () XksProxyAuthenticationCredential (Vous devez spécifier à la fois l'ID de clé d'accès et la clé d'accès secrète, même si vous ne modifiez qu'un seul élément.)	✓	
Chemin de l'URI du proxy (XksProxyUriPath)	✓	
Connectivité proxy (XksProxyConnectivity) (Vous devez également mettre à jour le point de terminaison d'URI de proxy. Si vous passez à la connectivité au service de point de terminaison d'un VPC, vous devez spécifier un nom de service de point de terminaison d'un VPC proxy.)		✓
Point de terminaison URI du proxy (XksProxyUriEndpoint)		✓

Propriétés du magasin de clés externe modifiables	Tout état de connexion	Exiger l'état Déconnecté
Si vous modifiez l'URI du point de terminaison du proxy, vous devrez peut-être aussi modifier le certificat TLS associé.		
Nom du service de point de terminaison VPC proxy () XksProxyVpcEndpointServiceName (Ce champ est obligatoire pour la connectivité au service de point de terminaison d'un VPC)		✓

Rubriques

- Modifier un magasin de clés externe (console)
- Modifier un magasin de clés externe (API)

Modifier un magasin de clés externe (console)

Lorsque vous modifiez un magasin de clés, vous pouvez changer n'importe laquelle des valeurs modifiables. Certaines modifications nécessitent que le magasin de clés externe soit déconnecté de son proxy de magasin de clés externe.

Si vous modifiez le chemin d'URI de proxy ou les informations d'identification pour l'authentification du proxy, vous pouvez saisir les nouvelles valeurs ou charger un fichier de configuration de proxy de magasin de clés externe qui contient les nouvelles valeurs.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Choisissez la ligne du magasin de clés externe que vous souhaitez modifier.

5. Si nécessaire, déconnectez le magasin de clés externe de son proxy de magasin de clés externe. Dans le menu Key store actions (Actions de magasin de clés), choisissez Disconnect (Déconnecter).
6. À partir du menu Key store actions (Actions de magasin de clés), choisissez Edit (Modifier).
7. Modifiez une ou plusieurs propriétés modifiables du magasin de clés externe. Vous pouvez également charger un fichier de configuration de proxy de magasin de clés externe contenant des valeurs pour le chemin d'URI de proxy et les informations d'identification pour l'authentification du proxy. Vous pouvez utiliser un fichier de configuration de proxy même si certaines valeurs spécifiées dans le fichier n'ont pas changé.
8. Choisissez Update external key store (Mettre à jour le magasin de clés externe).
9. Passez en revue l'avertissement et, si vous décidez de continuer, confirmez-le, puis choisissez Update external key store (Mettre à jour le magasin de clés externe).

   Lorsque la procédure se déroule avec succès, un message décrit les propriétés que vous avez modifiées. Si elle ne réussit pas, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre.
10. Si nécessaire, reconnectez le magasin de clés externe. Dans le menu Key store actions (Actions de magasin de clés), choisissez Connect (Connecter).

    Vous pouvez laisser le magasin de clés externe déconnecté. Mais, tant qu'il est déconnecté, vous ne pouvez pas créer de clés KMS dans le magasin de clés externe ou utiliser les clés KMS du magasin de clés externe pour les opérations cryptographiques.

Modifier un magasin de clés externe (API)

Pour modifier les propriétés d'un magasin de clés externe, utilisez l'UpdateCustomKeyStoreopération. Vous pouvez modifier plusieurs propriétés d'un magasin de clés externe dans la même opération. Si l'opération aboutit, AWS KMS renvoie une réponse HTTP 200 et un objet JSON sans propriétés.

Utilisez le paramètre `CustomKeyStoreId` pour identifier le magasin de clés externe. Utilisez les autres paramètres pour modifier les propriétés. Vous ne pouvez pas utiliser de fichier de configuration de proxy pour l'opération `UpdateCustomKeyStore`. Le fichier de configuration du proxy n'est pris en charge que par la console AWS KMS. Vous pouvez toutefois utiliser le fichier de configuration du proxy pour vous aider à déterminer les valeurs de paramètres correctes pour le proxy de votre magasin de clés externe.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Avant de commencer, si nécessaire, déconnectez le magasin de clés externe de son proxy de magasin de clés externe. Après la mise à jour, vous pouvez, si nécessaire, reconnecter le magasin de clés externe à son proxy de magasin de clés externe. Vous pouvez laisser le magasin de clés externe à l'état déconnecté, mais vous devez le reconnecter avant de pouvoir créer des clés KMS dans le magasin de clés ou d'utiliser les clés KMS existantes du magasin de clés pour les opérations cryptographiques.

> **Note**
>
> Si vous utilisez l'AWS CLI version 1.0, exécutez la commande suivante avant de spécifier un paramètre avec une valeur HTTP ou HTTPS, tel que le paramètre `XksProxyUriEndpoint`.
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> Sinon, l'AWS CLI version 1.0 remplace la valeur du paramètre par le contenu trouvé à cette adresse URI, provoquant l'erreur suivante :
>
> ```
> Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
> https:// : received non 200 status code of 404
> ```

Modifier le nom du magasin de clés externe

Le premier exemple utilise l'UpdateCustomKeyStoreopération pour changer le nom convivial du magasin de clés externe en`XksKeyStore`. La commande utilise le paramètre `CustomKeyStoreId` pour identifier le magasin de clés personnalisé et le paramètre `CustomKeyStoreName` pour spécifier le nouveau nom du magasin de clés personnalisé. Remplacez toutes les valeurs d'exemple par des valeurs réelles pour votre magasin de clés externe.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name XksKeyStore
```

Modifier les informations d'identification pour l'authentification du proxy

L'exemple suivant met à jour les informations d'identification pour l'authentification du proxy qu'AWS KMS utilise pour s'authentifier auprès du proxy de magasin de clés externe. Vous pouvez utiliser une

commande comme celle-ci pour effectuer une rotation des informations d'identification si elles ont subi une rotation sur votre proxy.

Mettez d'abord à jour les informations d'identification sur le proxy de votre magasin de clés externe. Utilisez ensuite cette fonctionnalité pour signaler la modification à AWS KMS. (Votre proxy prendra brièvement en charge les anciennes et les nouvelles informations d'identification afin que vous ayez le temps de les mettre à jour dans AWS KMS.)

Vous devez toujours spécifier à la fois l'ID de la clé d'accès et la clé d'accès secrète dans les informations d'identification, même si une seule valeur est modifiée.

Les deux premières commandes définissent des variables pour contenir les valeurs des informations d'identification. Les opérations `UpdateCustomKeyStore` utilisent le paramètre `CustomKeyStoreId` pour identifier le magasin de clés externe. Il utilise le paramètre `XksProxyAuthenticationCredential` avec ses champs `AccessKeyId` et `RawSecretAccessKey` pour spécifier les nouvelles informations d'identification. Remplacez toutes les valeurs d'exemple par des valeurs réelles pour votre magasin de clés externe.

```
$ accessKeyID=access key id
$ secretAccessKey=secret access key

$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
       --xks-proxy-authentication-credential \
           AccessKeyId=$accessKeyId,RawSecretAccessKey=$secretAccessKey
```

Modifier le chemin d'URI de proxy

L'exemple suivant met à jour le chemin d'URI de proxy (`XksProxyUriPath`). La combinaison du point de terminaison d'URI de proxy et du chemin d'URI de proxy doit être unique dans l'Compte AWS et la région. Remplacez toutes les valeurs d'exemple par des valeurs réelles pour votre magasin de clés externe.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-uri-path /kms/xks/v1
```

Modifier la connectivité au service de point de terminaison d'un VPC

L'exemple suivant utilise l'UpdateCustomKeyStoreopération pour modifier le type de connectivité du proxy du magasin de clés externe en`VPC_ENDPOINT_SERVICE`. Pour effectuer cette modification, vous devez spécifier les valeurs requises pour la connectivité au service de point

de terminaison d'un VPC, notamment le nom du service de point de terminaison d'un VPC (`XksProxyVpcEndpointServiceName`) et une valeur de point de terminaison d'URI de proxy (`XksProxyUriEndpoint`) qui inclut le nom DNS privé du service de point de terminaison d'un VPC. Remplacez toutes les valeurs d'exemple par des valeurs réelles pour votre magasin de clés externe.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
            --xks-proxy-connectivity "VPC_ENDPOINT_SERVICE" \
            --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
            --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-
svc-example
```

Passer à une connectivité au point de terminaison public

L'exemple suivant remplace le type de connectivité du proxy de magasin de clés externe par `PUBLIC_ENDPOINT`. Lorsque vous effectuez cette modification, vous devez mettre à jour la valeur du point de terminaison de l'URI de proxy (`XksProxyUriEndpoint`). Remplacez toutes les valeurs d'exemple par des valeurs réelles pour votre magasin de clés externe.

> **Note**
>
> La connectivité au point de terminaison d'un VPC offre une plus grande sécurité que la connectivité au point de terminaison public. Avant de passer à la connectivité au point de terminaison public, envisagez d'autres options, notamment la localisation de votre proxy de magasin de clés externe sur site et l'utilisation du VPC uniquement pour la communication.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
            --xks-proxy-connectivity "PUBLIC_ENDPOINT" \
            --xks-proxy-uri-endpoint https://myproxy.xks.example.com
```

Afficher un magasin de clés externe

Vous pouvez consulter les banques de clés externes de chaque compte et de chaque région à l'aide de la AWS KMS console ou à l'aide de l'DescribeCustomKeyStoresopération.

Lorsque vous consultez un magasin de clés externe, vous pouvez voir les éléments suivants :

- Des informations de base sur le magasin de clés, notamment son nom convivial, son ID, son type de magasin de clés et sa date de création.

- Des informations de configuration pour le proxy de magasin de clés externe, notamment le type de connectivité, le point de terminaison et le chemin d'URI de proxy, ainsi que l'ID de clé d'accès de vos informations d'identification pour l'authentification du proxy actuelles.
- Si le proxy de magasin de clés externe utilise la connectivité au service de point de terminaison d'un VPC, la console affiche le nom du service de point de terminaison d'un VPC.
- L'état de la connexion actuel.

> **Note**
>
> La valeur d'état de connexion Disconnected (Déconnectée) indique que le magasin de clés externe n'a jamais été connecté ou qu'il a été intentionnellement déconnecté de son proxy de magasin de clés externe. Cependant, si vos tentatives d'utiliser une clé KMS dans un magasin de clés externe connecté échouent, cela peut signifier la présence d'un problème avec le magasin de clés externe ou son proxy. Pour obtenir de l'aide, veuillez consulter Erreurs de connexion au magasin de clés externe.

- Une section de surveillance contenant des graphiques des CloudWatch statistiques Amazon conçues pour vous aider à détecter et à résoudre les problèmes liés à votre magasin de clés externe. Pour obtenir de l'aide pour interpréter les graphiques, les utiliser dans le cadre de votre planification et de votre résolution des problèmes, et pour créer des CloudWatch alarmes en fonction des indicateurs présentés dans les graphiques, consultezSurveiller un magasin de clés externe.

Voir aussi :

- Afficher des clés KMS dans un magasin de clés externe
- Journalisation des appels d'API AWS KMS avec AWS CloudTrail

Rubriques

- Propriétés du magasin de clés externe
- Consulter un magasin de clés externe (console)
- Consulter un magasin de clés externe (API)

Propriétés du magasin de clés externe

Les propriétés suivantes d'un magasin de clés externe sont visibles dans la AWS KMS console et dans la DescribeCustomKeyStoresréponse.

Propriétés du magasin de clés personnalisé

Les valeurs suivantes apparaissent dans la section General configuration (Configuration générale) de la page détaillée de chaque magasin de clés personnalisé. Ces propriétés s'appliquent à tous les magasins de clés personnalisés, y compris les magasins de clés AWS CloudHSM et les magasins de clés externes.

ID du magasin de clés personnalisé

Un ID unique qu'AWS KMS attribue au magasin de clés personnalisé.

Nom du magasin de clés personnalisé

Un nom convivial que vous attribuez au magasin de clés personnalisé lorsque vous le créez. Vous pouvez modifier cette valeur à tout moment.

Type de magasin de clés personnalisé

Le type de magasin de clés personnalisé. Les valeurs valides sont AWS CloudHSM (`AWS_CLOUDHSM`) ou Magasin de clés externe (`EXTERNAL_KEY_STORE`). Vous ne pouvez pas modifier le type après avoir créé le magasin de clés personnalisé.

Date de création

La date à laquelle le magasin de clés personnalisé a été créé. Cette date est affichée en heure locale pour l'Région AWS.

État de connexion

Indique si le magasin de clés personnalisé est connecté au magasin de clés de sauvegarde. L'état de connexion est `DISCONNECTED` uniquement si le magasin de clés personnalisé n'a jamais été connecté à son magasin de clés de sauvegarde, ou s'il a été déconnecté intentionnellement. Pour plus de détails, consultez the section called “État de connexion”.

Propriétés de configuration du magasin de clés externe

Les valeurs suivantes apparaissent dans la section Configuration du proxy du magasin de clés externe de la page détaillée de chaque magasin de clés externe et dans

l'`XksProxyConfiguration`élément de DescribeCustomKeyStoresréponse. Pour obtenir une description détaillée de chaque champ, y compris les exigences d'unicité et de l'aide pour déterminer la valeur correcte de chaque champ, veuillez consulter the section called “Rassembler les conditions requises” dans la rubrique Créer un magasin de clés externe.

Connectivité de proxy

Indique si le magasin de clés externe utilise une connectivité au point de terminaison public ou une connectivité au service de point de terminaison d'un VPC.

Point de terminaison d'URI de proxy

Le point de terminaison qu'AWS KMS utilise pour se connecter à votre proxy de magasin de clés externe.

Chemin d'URI de proxy

Le chemin depuis le point de terminaison d'URI de proxy où AWS KMS envoie les requêtes d'API du proxy.

Informations d'identification du proxy : ID de la clé d'accès

Fait partie des informations d'identification pour l'authentification du proxy que vous définissez sur votre proxy de magasin de clés externe. L'ID de clé d'accès identifie la clé d'accès secrète dans les informations d'identification.

AWS KMS utilise le processus de signature SigV4 et les informations d'identification pour l'authentification du proxy pour signer ses requêtes auprès de votre proxy de magasin de clés externe. Les informations d'identification dans la signature permettent au proxy du magasin de clés externe d'authentifier les requêtes en votre nom en provenance d'AWS KMS.

Nom du service de point de terminaison d'un VPC

Nom du service de point de terminaison d'un Amazon VPC prenant en charge votre magasin de clés externe. Cette valeur n'apparaît que lorsque le magasin de clés externe utilise la connectivité au service de point de terminaison d'un VPC. Vous pouvez localiser votre proxy de magasin de clés externe dans le VPC ou utiliser le service de point de terminaison d'un VPC pour communiquer en toute sécurité avec votre proxy de magasin de clés externe.

Consulter un magasin de clés externe (console)

Pour consulter les magasins de clés externes d'un compte et d'une région donnés, utilisez la procédure suivante.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Pour consulter des informations détaillées sur un magasin de clés externe, sélectionnez le nom du magasin de clés.

Consulter un magasin de clés externe (API)

Pour afficher vos stockages de clés externes, utilisez l'DescribeCustomKeyStoresopération. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Toutefois, vous pouvez utiliser le paramètre `CustomKeyStoreName` ou `CustomKeyStoreId` (mais pas les deux) pour limiter la sortie à un magasin de clés personnalisé en particulier.

Pour les magasins de clés personnalisées, la sortie contient l'ID, le nom et le type du magasin de clés personnalisé, ainsi que l'état de connexion du magasin de clés. Si l'état de connexion est `FAILED`, la sortie contient également un `ConnectionErrorCode` qui décrit la raison de l'erreur. Pour obtenir de l'aide pour interpréter le `ConnectionErrorCode` pour un magasin de clés externe, veuillez consulter la rubrique Codes d'erreur de connexion pour les magasins de clés externes.

Pour les magasins de clés externes, la sortie contient également l'élément `XksProxyConfiguration`. Cet élément inclut le type de connectivité, le point de terminaison d'URI de proxy, le chemin d'URI de proxy et l'ID de clé d'accès des informations d'identification pour l'authentification du proxy.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Par exemple, la commande suivante renvoie tous les magasins de clés personnalisées du compte et de la région. Vous pouvez utiliser les paramètres `Marker` et `Limit` pour parcourir les magasins de clés personnalisés de la sortie.

```
$ aws kms describe-custom-key-stores
```

La commande suivante utilise le paramètre `CustomKeyStoreName` pour obtenir uniquement l'exemple de magasin de clés externe avec le nom convivial `ExampleXksPublic`. Cet exemple de

magasin de clés utilise la connectivité au point de terminaison public. Il est connecté à son proxy de magasin de clés externe.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksPublic
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleXksPublic",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-14T20:17:36.419000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://xks.example.com:6443",
        "UriPath": "/example/prefix/kms/xks/v1"
      }
    }
  ]
}
```

La commande suivante permet d'obtenir un exemple de magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC. Dans cet exemple, le magasin de clés externe est connecté à son proxy de magasin de clés externe.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
```

```
  ]
}
```

Un `ConnectionState` dont la valeur est `Disconnected` indique que le magasin de clés externe n'a jamais été connecté ou qu'il a été intentionnellement déconnecté de son proxy de magasin de clés externe. Cependant, si les tentatives d'utilisation d'une clé KMS dans un magasin de clés externe connecté échouent, cela peut indiquer un problème avec le proxy du magasin de clés externe ou avec d'autres composants externes.

Si le `ConnectionState` du magasin de clés externe est `FAILED`, la réponse de `DescribeCustomKeyStores` inclut un élément `ConnectionErrorCode` qui explique la raison de l'erreur.

Par exemple, dans la sortie suivante, la valeur `XKS_PROXY_TIMED_OUT` indique qu'AWS KMS peut se connecter au proxy de magasin de clés externe, mais que la connexion a échoué, car le proxy de magasin de clés externe n'a pas répondu à AWS KMS dans le délai imparti. Si ce code d'erreur de connexion s'affiche à plusieurs reprises, informez-en le fournisseur du proxy de votre magasin de clés externe. Pour obtenir de l'aide sur ce sujet et sur d'autres échecs de connexion, consultez Résoudre les problèmes liés aux magasins de clés externes.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_PROXY_TIMED_OUT",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Surveiller un magasin de clés externe

AWS KMScollecte des statistiques pour chaque interaction avec un magasin de clés externe et les publie sur votre CloudWatch compte. Ces métriques sont utilisées pour générer les graphiques dans la section de surveillance de la page détaillée pour chaque magasin de clés externe. La rubrique suivante explique comment utiliser les graphiques pour identifier et résoudre les problèmes de fonctionnement et de configuration affectant votre magasin de clés externe. Nous vous recommandons d'utiliser les CloudWatch métriques pour définir des alarmes qui vous avertissent lorsque votre magasin de clés externe ne fonctionne pas comme prévu. Pour plus d'informations, consultez la section Surveillance avec Amazon CloudWatch.

Rubriques

- Afficher les graphiques
- Interpréter les graphiques
- Définition d'alarmes

Afficher les graphiques

Vous pouvez afficher les graphiques dans différents niveaux de détails. Par défaut, chaque graphique utilise une plage de temps de trois heures et une période d'agrégation de cinq minutes. Vous pouvez ajuster l'affichage graphique dans la console, mais vos modifications reviendront aux paramètres par défaut lorsque la page détaillée du magasin de clés externe sera fermée ou que le navigateur sera actualisé. Pour obtenir de l'aide sur CloudWatch la terminologie Amazon, consultez Amazon CloudWatch Concepts.

Afficher les détails des points de données

Les données de chaque graphique sont collectées par les métriques AWS KMS. Pour afficher plus d'informations sur un point de données spécifique, placez le pointeur de la souris sur le point de données du graphique linéaire. Cela affichera une fenêtre contextuelle contenant plus d'informations sur la métrique dont le graphique est issu. Chaque élément de la liste affiche la valeur de dimension enregistrée à ce point de données. La fenêtre contextuelle affiche une valeur nulle (–) si aucune donnée de métrique n'est disponible pour la valeur de dimension à ce point de données. Certains graphiques enregistrent plusieurs dimensions et valeurs pour un seul point de données. D'autres graphiques, tels que le graphique de fiabilité, utilisent les données collectées par la métrique pour calculer une valeur unique. Chaque élément de la liste est associé à une couleur de graphique linéaire différente.

Modifier la plage de temps

Pour modifier la plage de temps, sélectionnez l'une des plages de temps prédéfinies dans le coin supérieur droit de la section de surveillance. Les plages de temps prédéfinies s'étendent de 1 heure à 1 semaine (1 h, 3 h, 12 h, 1 j, 3 j, ou 1 sem). Cela permet d'ajuster la plage de temps pour tous les graphiques. Si vous souhaitez afficher un graphique spécifique dans un intervalle de temps différent, ou si vous souhaitez définir un intervalle de temps personnalisé, agrandissez-le ou affichez-le dans la CloudWatch console Amazon.

Zoom avant sur les graphiques

Vous pouvez utiliser la fonctionnalité de zoom de la mini-carte pour vous concentrer sur des sections de graphiques linéaires et des parties empilées des graphiques sans basculer entre les vues zoomée et dézoomée. Par exemple, vous pouvez utiliser la fonctionnalité de zoom de la mini-carte pour mettre l'accent sur un pic dans un graphique, de sorte que vous puissiez comparer le pic à d'autres graphiques de la section de surveillance provenant de la même chronologie.

1. Choisissez et faites glisser la zone du graphique sur laquelle vous souhaitez mettre l'accent, puis déposez.
2. Pour réinitialiser le zoom, choisissez l'icône Reset zoom (Réinitialiser le zoom), qui ressemble à une loupe avec un symbole moins (-) à l'intérieur.

Agrandir un graphique

Pour agrandir un graphique, sélectionnez l'icône de menu dans le coin supérieur droit d'un graphique individuel et choisissez Enlarge (Agrandir). Vous pouvez également sélectionner l'icône d'agrandissement qui apparaît à côté de l'icône de menu lorsque vous passez la souris sur un graphique.

L'agrandissement d'un graphique vous permet de modifier davantage son affichage en spécifiant une période, une plage de temps personnalisée ou un intervalle d'actualisation différents. Ces modifications reviendront aux paramètres par défaut lorsque vous fermerez la vue agrandie.

Modifier la période

1. Choisissez le menu Period options (Options de période). Par défaut, ce menu affiche la valeur : 5 minutes.
2. Choisissez une période, les périodes prédéfinies s'étendent de 1 seconde à 30 jours.

Par exemple, vous pouvez choisir une vue d'une minute, ce qui peut être utile lors d'un dépannage. Vous pouvez également choisir une vue moins détaillée, d'une heure par exemple. Cela peut être utile lors de l'affichage d'une plage de temps plus large (par exemple, 3 jours), afin de voir les tendances au fil du temps. Pour plus d'informations, consultez la section Périodes dans le guide de CloudWatch l'utilisateur Amazon.

Modifier la plage de temps ou le fuseau horaire

1. Sélectionnez l'une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1 h, 3 h, 12 h, 1 j, 3 j, ou 1 sem). Vous pouvez également choisir Custom (Personnalisée) pour définir votre propre plage horaire.
2. Choisissez Custom (Personnalisée).
   a. Plage de temps : sélectionnez l'onglet Absolute (Absolue) dans le coin supérieur gauche de la boîte de dialogue. Utilisez le sélecteur de calendrier ou les champs de texte pour spécifier la plage de temps.
   b. Fuseau horaire : choisissez le menu déroulant dans le coin supérieur droit de la boîte de dialogue. Vous pouvez changer le fuseau horaire sur UTC ou Local time zone (Fuseau horaire local).
3. Une fois que vous avez spécifié une plage de temps, choisissez Apply (Appliquer).

Modifier la fréquence à laquelle les données de votre graphique sont actualisées

1. Dans le coin supérieur droit, choisissez le menu Refresh options (Options d'actualisation).
2. Choisissez un intervalle d'actualisation (Désactivé, 10 secondes, 1 minute, 2 minutes, 5 minutes ou 15 minutes).

Afficher les graphiques dans la CloudWatch console Amazon

Les graphiques de la section de surveillance sont dérivés de mesures prédéfinies AWS KMS publiées sur Amazon CloudWatch. Vous pouvez les ouvrir dans la CloudWatch console et les enregistrer dans des CloudWatch tableaux de bord. Si vous possédez plusieurs magasins de clés externes, vous pouvez ouvrir leurs graphiques respectifs CloudWatch et les enregistrer dans un tableau de bord unique pour comparer leur état de santé et leur utilisation.

Ajouter au CloudWatch tableau de bord

Sélectionnez Ajouter au tableau de bord dans le coin supérieur droit pour ajouter tous les graphiques à un tableau de CloudWatch bord Amazon. Vous pouvez utiliser un tableau de bord existant ou en créer un. Pour plus d'informations sur l'utilisation de ce tableau de bord pour créer des vues personnalisées des graphiques et des alarmes, consultez la section Utilisation CloudWatch des tableaux de bord Amazon dans le guide de CloudWatch l'utilisateur Amazon.

Afficher dans les CloudWatch métriques

Sélectionnez l'icône du menu dans le coin supérieur droit d'un graphique individuel et choisissez Afficher dans les métriques pour afficher ce graphique dans la CloudWatch console Amazon. Depuis la CloudWatch console, vous pouvez ajouter ce graphique unique à un tableau de bord et modifier les plages de temps, les périodes et les intervalles d'actualisation. Pour plus d'informations, consultez la section Représentation graphique des métriques dans le guide de CloudWatch l'utilisateur Amazon.

Interpréter les graphiques

AWS KMS fournit plusieurs graphiques pour surveiller l'état de votre magasin de clés externe au sein de la console AWS KMS. Ces graphiques sont automatiquement configurés et dérivés des métriques AWS KMS.

Les données de graphique sont collectées dans le cadre des appels que vous effectuez vers votre magasin de clés externe et vos clés externes. Il se peut que des données apparaissent dans les graphiques pour une période pendant laquelle vous n'avez effectué aucun appel. Ces données proviennent des appels `GetHealthStatus` périodiques qu'AWS KMS effectue en votre nom pour vérifier l'état de votre proxy de magasin de clés externe et de votre gestionnaire de clés externe. Si vos graphiques affichent le message No data available (Aucune donnée disponible), cela signifie qu'aucun appel n'a été enregistré au cours de cette période ou que votre magasin de clés externe est à l'état `DISCONNECTED`. Vous pouvez peut-être identifier l'heure à laquelle votre magasin de clés externe s'est déconnecté en ajustant votre affichage sur une plage de temps plus étendue.

Rubriques

- Total requests (Nombre total de requêtes)
- Fiabilité
- Latence
- Les cinq principales exceptions
- Nombre de jours avant l'expiration du certificat

Total requests (Nombre total de requêtes)

Nombre total de requêtes AWS KMS reçues pour un magasin de clés externe spécifique au cours d'une plage de temps donnée. Utilisez ce graphique pour déterminer si vous êtes exposé à un risque de limitation.

AWS KMS recommande que votre gestionnaire de clés externe soit capable de traiter jusqu'à 1 800 requêtes d'opérations cryptographiques par seconde. Si vous approchez les 540 000 appels par période de cinq minutes, vous risquez d'être limité.

Vous pouvez contrôler le nombre de requêtes d'opérations cryptographiques sur des clés KMS dans votre magasin de clés externe qu'AWS KMS limite à l'aide de la métrique ExternalKeyStoreThrottle.

Si vous recevez des erreurs `KMSInvalidStateException` très fréquentes avec un message expliquant que la requête a été rejetée « en raison d'un taux de requêtes très élevé », cela peut indiquer que votre gestionnaire de clés externe ou votre proxy de magasin de clés externe ne peuvent pas suivre le rythme du taux de requêtes actuel. Si possible, réduisez votre taux de requêtes. Vous pouvez également envisager de demander une diminution de la valeur de votre quota de requêtes du magasin de clés personnalisé. Diminuer cette valeur de quota pourrait augmenter la limitation, mais cela indique qu'AWS KMS rejette rapidement les requêtes excédentaires avant qu'elles ne soient envoyées à votre proxy de magasin de clés externe ou à votre gestionnaire de clés externe. Pour solliciter une réduction de quota, accédez au Centre AWS Support et créez une demande.

Le graphique du nombre total de requêtes est dérivé de la métrique XksProxyErrors, qui collecte des données sur les réponses fructueuses et infructueuses qu'AWS KMS reçoit de votre proxy de magasin de clés externe. Lorsque vous consultez un point de données spécifique, la fenêtre contextuelle affiche la valeur de la dimension `CustomKeyStoreId` ainsi que le nombre total de requêtes AWS KMS enregistrées à ce point de données. Le `CustomKeyStoreId` sera toujours identique.

Fiabilité

Pourcentage de requêtes AWS KMS pour lesquelles le proxy de magasin de clés externe a renvoyé soit une réponse positive, soit une erreur non récupérable. Utilisez ce graphique pour évaluer l'état opérationnel de votre proxy de magasin de clés externe.

Lorsque le graphique affiche une valeur inférieure à 100 %, il indique les cas où le proxy n'a pas répondu ou a répondu par une erreur récupérable. Cela peut indiquer des problèmes liés au réseau,

une lenteur du proxy de magasin de clés externe ou du gestionnaire de clés externe, ou des bogues d'implémentation.

Si la requête inclut des informations d'identification erronées et que votre proxy répond par une exception `AuthenticationFailedException`, le graphique indiquera toujours une fiabilité de 100 %, car le proxy a identifié une valeur incorrecte dans la requête d'API de proxy de magasin de clés externe. Par conséquent, l'échec est prévisible. Si le pourcentage de votre graphique de fiabilité est de 100 %, cela signifie que le proxy de votre magasin de clés externe répond comme prévu. Si le graphique affiche une valeur inférieure à 100 %, le proxy a répondu par une erreur récupérable ou a expiré. Par exemple, si le proxy répond par une exception `ThrottlingException` en raison d'un taux de requêtes très élevé, il affichera un pourcentage de fiabilité inférieur, car le proxy n'a pas été en mesure d'identifier un problème spécifique dans la requête qui a provoqué son échec. En effet, les erreurs récupérables sont probablement des problèmes transitoires qui peuvent être résolus en répétant la requête.

Les réponses d'erreur suivantes réduiront le pourcentage de fiabilité. Vous pouvez utiliser le graphique Les cinq principales exceptions et la métrique XksProxyErrors pour surveiller davantage la fréquence à laquelle votre proxy renvoie chaque erreur récupérable.

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

Le graphique de fiabilité est dérivé de la métrique XksProxyErrors, qui collecte des données sur les réponses fructueuses et infructueuses qu'AWS KMS reçoit de votre proxy de magasin de clés externe. Le pourcentage de fiabilité ne diminue que si la réponse a une valeur `ErrorType` égale à `Retryable`. Lorsque vous consultez un point de données spécifique, la fenêtre contextuelle affiche la valeur de la dimension `CustomKeyStoreId` ainsi que le pourcentage de fiabilité des requêtes AWS KMS enregistrées à ce point de données. Le `CustomKeyStoreId` sera toujours identique.

Nous vous recommandons d'utiliser cette XksProxyErrors métrique pour créer une CloudWatch alarme qui vous avertit des problèmes potentiels liés au réseau en vous alertant lorsque plus de cinq erreurs réessayables sont enregistrées en une minute. Pour plus d'informations, consultez Création d'une CloudWatch alarme Amazon pour les erreurs réessayables.

Latence

Nombre de millisecondes nécessaires à un proxy de magasin de clés externe pour répondre à une requête AWS KMS. Utilisez ce graphique pour évaluer les performances de votre proxy de magasin de clés externe et de votre gestionnaire de clés externe.

AWS KMS s'attend à ce que le proxy de magasin de clés externe réponde à chaque requête dans un délai de 250 millisecondes. En cas de délai d'expiration du réseau, AWS KMS relancera la requête une seule fois. Si le proxy échoue une seconde fois, la latence enregistrée est le délai d'expiration combiné pour les deux tentatives de requête et le graphique affichera environ 500 millisecondes. Dans tous les autres cas où le proxy ne répond pas dans la limite du délai d'expiration de 250 millisecondes, la latence enregistrée est de 250 millisecondes. Si le proxy expire fréquemment lors des opérations de chiffrement et de déchiffrement, consultez votre administrateur proxy externe. Pour obtenir de l'aide afin de résoudre les problèmes de latence, veuillez consulter la rubrique Erreurs de latence et de délai d'expiration.

Des réponses lentes peuvent également indiquer que votre gestionnaire de clés externe ne peut pas gérer le trafic de requête actuel. AWS KMS recommande que votre gestionnaire de clés externe soit capable de traiter jusqu'à 1 800 requêtes d'opérations cryptographiques par seconde. Si votre gestionnaire de clés externe ne peut pas gérer le taux de 1 800 requêtes par seconde, pensez à demander une diminution de votre quota de requêtes de clés KMS dans un magasin de clés personnalisé. Les requêtes d'opérations cryptographiques utilisant les clés KMS de votre magasin de clés externe échoueront rapidement, avec une exception de limitation, au lieu d'être traitées puis rejetées par le proxy de votre magasin de clés externe ou le gestionnaire de clés externe.

Le graphique de latence est dérivé de la métrique XksProxyLatency. Lorsque vous consultez un point de données spécifique, la fenêtre contextuelle affiche les valeurs des dimensions `KmsOperation` et `XksOperation` correspondantes, ainsi que la latence moyenne enregistrée pour les opérations sur ce point de données. Les éléments de la liste sont classés de la latence la plus élevée à la plus faible.

Nous vous recommandons d'utiliser cette XksProxyLatency métrique pour créer une CloudWatch alarme qui vous avertira lorsque votre latence approche de la limite de temporisation. Pour plus d'informations, consultez Création d'une CloudWatch alarme Amazon pour l'expiration du délai de réponse.

Les cinq principales exceptions

Les cinq principales exceptions en cas d'échec des opérations cryptographiques et de gestion au cours d'une période donnée. Utilisez ce graphique pour suivre les erreurs les plus fréquentes, afin de prioriser votre effort d'ingénierie.

Ce nombre inclut les exceptions qu'AWS KMS reçoit du proxy de magasin de clés externe et les `XksProxyUnreachableException` qu'AWS KMS renvoie en interne lorsqu'il ne peut pas établir de communication avec le proxy de magasin de clés externe.

Des taux élevés d'erreurs récupérables peuvent indiquer des erreurs réseau, tandis que des taux élevés d'erreurs non récupérables peuvent indiquer un problème de configuration de votre magasin de clés externe. Par exemple, un pic d'exceptions `AuthenticationFailedExceptions` indique une différence entre les informations d'identification pour l'authentification configurées dans AWS KMS et le proxy de magasin de clés externe. Pour consulter la configuration de votre magasin de clés externe, veuillez consulter la rubrique Afficher un magasin de clés externe. Pour modifier les paramètres de votre clé externe, veuillez consulter la rubrique Modifier les propriétés du magasin de clés externe.

Les exceptions que AWS KMS reçoit du proxy du magasin de clés externes sont différentes des exceptions que AWS KMS renvoie lorsqu'une opération échoue. Les opérations de chiffrement AWS KMS renvoient une exception `KMSInvalidStateException` pour tous les échecs liés à la configuration externe ou à l'état de connexion du magasin de clés externes. Pour identifier le problème, utilisez le texte du message d'erreur qui l'accompagne.

Le tableau suivant montre les exceptions qui peuvent apparaître dans le graphique des cinq principales exceptions et les exceptions correspondantes qu'AWS KMS vous renvoie.

Error type (Type d'erreur)	Exception affichée dans le graphique	Exception qu'AWS KMS vous a renvoyée
Non récupérable	**`AccessDeniedException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Problèmes d'autorisation du proxy.	**`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.

Error type (Type d'erreur)	Exception affichée dans le graphique	Exception qu'AWS KMS vous a renvoyée
Non récupérable	**`AuthenticationFailedException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs liées aux informations d'identification pour l'authentification.	**`XksProxyIncorrectAuthenticationCredentialException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Récupérable	**`DependencyTimeoutException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs de latence et de délai d'expiration.	**`XksProxyUriUnreachableException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.

Error type (Type d'erreur)	Exception affichée dans le graphique	Exception qu'AWS KMS vous a renvoyée
Récupérable	**`InternalException`** Le proxy de magasin de clés externe a rejeté la requête, car il ne peut pas communiquer avec le gestionnaire de clés externe. Vérifiez que la configuration du proxy de magasin de clés externe est correcte et que le gestionnaire de clés externe est disponible.	**`XksProxyInvalidResponseException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Non récupérable	**`InvalidCiphertextException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs de déchiffrement.	**`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Non récupérable	**`InvalidKeyUsageException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs d'opérations cryptographiques pour la clé externe.	**`XksKeyInvalidConfigurationException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.

Error type (Type d'erreur)	Exception affichée dans le graphique	Exception qu'AWS KMS vous a renvoyée
Non récupérable	**`InvalidStateException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs d'opérations cryptographiques pour la clé externe.	**`XksKeyInvalidConfigurationException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Non récupérable	**`InvalidUriPathException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs de configuration générale.	**`XksProxyInvalidConfigurationException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Non récupérable	**`KeyNotFoundException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs liées aux clés externes.	**`XksKeyNotFoundException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.

Error type (Type d'erreur)	Exception affichée dans le graphique	Exception qu'AWS KMS vous a renvoyée
Récupérable	**`ThrottlingException`** Le proxy de magasin de clés externe a rejeté la requête en raison d'un taux de requêtes très élevé. Réduisez la fréquence de vos appels utilisant des clés KMS dans ce magasin de clés externe.	**`XksProxyUriUnreachableException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Non récupérable	**`UnsupportedOperationException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Erreurs d'opérations cryptographiques pour la clé externe.	**`XksKeyInvalidResponseException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.

Error type (Type d'erreur)	Exception affichée dans le graphique	Exception qu'AWS KMS vous a renvoyée
Non récupérable	**`ValidationException`** Pour bénéficier d'une aide à la résolution des problèmes, consultez Problèmes liés aux proxys.	**`XksProxyInvalidResponseException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.
Récupérable	**`XksProxyUnreachableException`** Si cette erreur s'affiche à plusieurs reprises, vérifiez que le proxy de votre magasin de clés externe est actif et connecté au réseau, et que son chemin d'URI et son URI de point de terminaison ou le nom de service VPC sont corrects dans votre magasin de clés externe.	**`XksProxyUriUnreachableException`** dans la réponse aux opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** dans la réponse aux opérations `CreateKey` . **`KMSInvalidStateException`** dans la réponse aux opérations cryptographiques.

Le graphique des cinq principales exceptions est dérivé de la métrique XksProxyErrors. Lorsque vous consultez un point de données spécifique, la fenêtre contextuelle affiche la valeur de la

dimension `ExceptionName` ainsi que le nombre de fois que l'exception a été enregistrée à ce point de données. Les cinq éléments de la liste sont classés de l'exception la plus fréquente à la moins fréquente.

Nous vous recommandons d'utiliser cette XksProxyErrors métrique pour créer une CloudWatch alarme qui vous avertit des problèmes de configuration potentiels en vous alertant lorsque plus de cinq erreurs non réessayables sont enregistrées en une minute. Pour plus d'informations, consultez Création d'une CloudWatch alarme Amazon pour les erreurs non réessayables.

Nombre de jours avant l'expiration du certificat

Nombre de jours avant l'expiration du certificat TLS de votre point de terminaison du proxy de magasin de clés externe (`XksProxyUriEndpoint`). Utilisez ce graphique pour surveiller l'expiration imminente de votre certificat TLS.

Lorsque le certificat expire, AWS KMS ne peut plus communiquer avec le proxy de magasin de clés externe. Toutes les données protégées par des clés KMS dans votre magasin de clés externe deviennent inaccessibles jusqu'à ce que vous renouveliez le certificat.

Le graphique du nombre de jours avant l'expiration du certificat est dérivé de la métrique XksProxyCertificateDaysToExpire. Nous vous recommandons vivement d'utiliser cette métrique pour créer une CloudWatch alarme qui vous avertira de l'expiration prochaine. L'expiration du certificat peut vous empêcher d'accéder à vos ressources chiffrées. Réglez l'alerte pour donner à votre organisation le temps de renouveler le certificat avant qu'il n'expire. Pour plus d'informations, consultez Création d'une CloudWatch alarme Amazon pour l'expiration du certificat.

Définition d'alarmes

Les graphiques de la section de surveillance fournissent une vue d'ensemble de l'état de vos magasins de clés externes et de vos clés KMS dans des magasins de clés externes pendant une période donnée. Cependant, vous pouvez créer des CloudWatch alarmes Amazon en fonction des statistiques externes du magasin clé afin de vous avertir lorsqu'une valeur de métrique dépasse le seuil que vous avez spécifié. L'alerte peut envoyer le message à une rubrique Amazon Simple Notification Service (Amazon SNS) ou à une politique Amazon EC2 Auto Scaling. Pour obtenir des informations détaillées sur les CloudWatch alarmes, consultez la section Utilisation des CloudWatch alarmes Amazon dans le guide de CloudWatch l'utilisateur Amazon.

Avant de créer une CloudWatch alarme Amazon, vous avez besoin d'une rubrique Amazon SNS. Pour plus de détails, consultez la rubrique Création d'un Amazon SNS dans le guide de CloudWatch l'utilisateur Amazon.

Rubriques

- Création d'une CloudWatch alarme Amazon pour l'expiration du certificat
- Création d'une CloudWatch alarme Amazon pour l'expiration du délai de réponse
- Création d'une CloudWatch alarme Amazon pour les erreurs réessayables
- Création d'une CloudWatch alarme Amazon pour les erreurs non réessayables

Création d'une CloudWatch alarme Amazon pour l'expiration du certificat

Cette alarme utilise la XksProxyCertificateDaysToExpire métrique AWS KMS publiée sur CloudWatch pour enregistrer l'expiration prévue du certificat TLS associé à votre point de terminaison proxy de stockage de clés externe. Vous ne pouvez pas créer une alerte unique pour tous les magasins de clés externes de votre compte ou une alerte pour les magasins de clés externes que vous pourriez créer à l'avenir.

Nous vous recommandons de régler l'alerte pour qu'elle vous avertisse 10 jours avant l'expiration de votre certificat, mais vous êtes invité à définir le seuil qui correspond le mieux à vos besoins.

Créez l'alarme

Suivez les instructions de la section Création CloudWatch d'une alarme basée sur un seuil statique en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Sélectionner une métrique	Choisissez KMS, puis choisissez XKS Proxy Certificate Metrics (Métriques de certificat de proxy XKS). Cochez la case à côté du `XksProxyCertificateName` que vous souhaitez surveiller. Ensuite, choisissez Select metric (Sélectionner une métrique).
Statistique	Minimum
Période	5 minutes
Type de seuil	Statique

Champ	Valeur
Chaque fois que …	À chaque fois `Lower` que XksProxyCertificateDaysToExpirec'est le cas`10`.

Création d'une CloudWatch alarme Amazon pour l'expiration du délai de réponse

Cette alarme utilise la XksProxyLatency métrique AWS KMS publiée sur CloudWatch pour enregistrer le nombre de millisecondes nécessaires à un proxy de stockage de clés externe pour répondre à une demande. AWS KMS Vous ne pouvez pas créer une alerte unique pour tous les magasins de clés externes de votre compte ou une alerte pour les magasins de clés externes que vous pourriez créer à l'avenir.

AWS KMS s'attend à ce que le proxy de magasin de clés externe réponde à chaque requête dans un délai de 250 millisecondes. Nous vous recommandons de configurer une alerte pour vous avertir lorsque le proxy de votre magasin de clés externe met plus de 200 millisecondes à répondre, mais vous êtes invité à définir le seuil qui correspond le mieux à vos besoins.

Créez l'alarme

Suivez les instructions de la section Création CloudWatch d'une alarme basée sur un seuil statique en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Sélectionner une métrique	Choisissez KMS, puis choisissez XKS Proxy Latency Metrics (Métriques de latence de proxy XKS). Cochez la case à côté du `KmsOperation` que vous souhaitez surveiller. Ensuite, choisissez Select metric (Sélectionner une métrique).
Statistique	Moyenne
Période	5 minutes
Type de seuil	Statique

Champ	Valeur
Chaque fois que …	À chaque fois `Greater` que XksProxyLatencyc'est le cas`200`.

Création d'une CloudWatch alarme Amazon pour les erreurs réessayables

Cette alarme utilise la XksProxyErrors métrique AWS KMS publiée sur CloudWatch pour enregistrer le nombre d'exceptions liées aux AWS KMS demandes adressées à votre proxy de stockage de clés externe. Vous ne pouvez pas créer une alerte unique pour tous les magasins de clés externes de votre compte ou une alerte pour les magasins de clés externes que vous pourriez créer à l'avenir.

Les erreurs récupérables réduisent votre pourcentage de fiabilité et peuvent indiquer des erreurs réseau. Nous vous recommandons de définir une alerte pour vous avertir lorsque plus de cinq erreurs récupérables sont enregistrées sur une période d'une minute, mais vous êtes invité à définir le seuil qui correspond le mieux à vos besoins.

Suivez les instructions de la section Création CloudWatch d'une alarme basée sur un seuil statique en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Sélectionner une métrique	Choisissez l'onglet Query (Requête). Choisissez `AWS/KMS` comme Namespace (Espace de noms). Saisissez `SUM(XksProxyErrors)` comme Metric name (Nom de la métrique). Saisissez `ErrorType = Retryable` pour Filter by (Filtrer par). Cliquez sur Exécuter. Ensuite, choisissez Select metric (Sélectionner une métrique).
Étiquette	*`Erreurs récupérables`*
Période	1 minute

Champ	Valeur
Type de seuil	Statique
Chaque fois que …	Chaque fois que `q1` est `Greater` que `5`.

Création d'une CloudWatch alarme Amazon pour les erreurs non réessayables

Cette alarme utilise la XksProxyErrors métrique AWS KMS publiée sur CloudWatch pour enregistrer le nombre d'exceptions liées aux AWS KMS demandes adressées à votre proxy de stockage de clés externe. Vous ne pouvez pas créer une alerte unique pour tous les magasins de clés externes de votre compte ou une alerte pour les magasins de clés externes que vous pourriez créer à l'avenir.

Les erreurs non récupérables peuvent indiquer un problème de configuration de votre magasin de clés externe. Nous vous recommandons de définir une alerte pour vous avertir lorsque plus de cinq erreurs non récupérables sont enregistrées sur une période d'une minute, mais vous êtes invité à définir le seuil qui correspond le mieux à vos besoins.

Suivez les instructions de la section Création CloudWatch d'une alarme basée sur un seuil statique en utilisant les valeurs obligatoires suivantes. Pour les autres champs, acceptez les valeurs par défaut et fournissez les noms demandés.

Champ	Valeur
Sélectionner une métrique	Choisissez l'onglet Query (Requête). Choisissez `AWS/KMS` comme Namespace (Espace de noms). Saisissez `SUM(XksProxyErrors)` comme Metric name (Nom de la métrique). Saisissez `ErrorType = Non-retryable` pour Filter by (Filtrer par). Cliquez sur Exécuter. Ensuite, choisissez Select metric (Sélectionner une métrique).
Étiquette	*Erreurs non récupérables*

Champ	Valeur
Période	1 minute
Type de seuil	Statique
Chaque fois que …	Chaque fois que q1 est `Greater` que 5.

Connecter et déconnecter un magasin de clés externe

Les nouveaux magasins de clés externes ne sont pas connectés. Pour créer et utiliser des AWS KMS keys dans votre magasin de clés externe, vous devez connecter votre magasin de clés externe à son proxy de magasin de clés externe. Vous pouvez connecter et déconnecter votre magasin de clés externe à tout moment, et afficher son état de connexion.

Lorsque votre magasin de clés externe est déconnecté, AWS KMS ne peut pas communiquer avec votre proxy de magasin de clés externe. Par conséquent, vous pouvez afficher et gérer votre magasin de clés externe et ses clés KMS existantes. Toutefois, vous ne pouvez pas créer de clés KMS dans votre magasin de clés externe, ni utiliser ses clés KMS dans des opérations cryptographiques. Il se peut que vous deviez déconnecter votre magasin de clés externe à un moment donné, par exemple lorsque vous modifiez ses propriétés, mais planifiez cette action en conséquence. La déconnexion du magasin de clés peut perturber le fonctionnement des services AWS qui utilisent ses clés KMS.

Vous n'avez pas besoin de connecter votre magasin de clés externe. Vous pouvez conserver un magasin de clés externe dans un état déconnecté indéfiniment et le connecter uniquement lorsque vous avez besoin de l'utiliser. Cependant, vous pouvez tester la connexion régulièrement pour vérifier que les paramètres sont corrects et que le magasin peut être connecté.

Lorsque vous déconnectez un magasin de clés personnalisé, les clés KMS du magasin de clés deviennent immédiatement inutilisables (sous réserve d'une éventuelle cohérence). Toutefois, les ressources chiffrées à l'aide de clés de données protégées par la clé KMS ne sont pas affectées tant que la clé KMS n'est pas réutilisée, par exemple pour déchiffrer la clé de données. Ce problème affecte les Services AWS, dont beaucoup utilisent des clés de données pour protéger vos ressources. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

> **Note**
>
> Les magasins de clés externes sont à l'état `DISCONNECTED` uniquement lorsque le magasin de clés n'a jamais été connecté ou que vous le déconnectez explicitement. Un état `CONNECTED` n'indique pas que le magasin de clés externe ou ses composants de support fonctionnent efficacement. Pour plus d'informations sur les performances des composants de votre magasin de clés externe, veuillez consulter les graphiques de la section Monitoring (Surveillance) de la page détaillée de chaque magasin de clés externe. Pour plus de détails, consultez Surveiller un magasin de clés externe.
>
> Votre gestionnaire de clés externe peut fournir des méthodes supplémentaires pour arrêter et redémarrer la communication entre votre magasin de clés externe AWS KMS et votre proxy de magasin de clés externe, ou entre votre proxy de magasin de clés externe et le gestionnaire de clés externe. Pour en savoir plus, veuillez consulter la documentation de votre gestionnaire de clés externe.

Rubriques

- Connecter un magasin de clés externe
- Déconnexion d'un magasin de clés externe
- État de connexion
- Connecter un magasin de clés externe (console)
- Connecter un magasin de clés externe (API)
- Déconnecter un magasin de clés externe (console)
- Déconnecter un magasin de clés externe (API)

Connecter un magasin de clés externe

Lorsque votre magasin de clés externe est connecté à son proxy de magasin de clés externe, vous pouvez créer des clés KMS dans votre magasin de clés externe et utiliser les clés KMS existantes dans les opérations cryptographiques.

Le processus qui connecte un magasin de clés externe à son proxy de magasin de clés externe varie en fonction de la connectivité du magasin de clés externe.

- Lorsque vous connectez un magasin de clés externe connecté à un point de terminaison public, AWS KMS envoie une GetHealthStatus demande au proxy du magasin de clés externe pour

valider le point de terminaison de l'URI du proxy, le chemin de l'URI du proxy et les informations d'authentification du proxy. Une réponse positive du proxy confirme que le point de terminaison d'URI de proxy et le chemin d'URI de proxy sont exacts et accessibles, et que le proxy a authentifié la requête signée à l'aide des informations d'identification pour l'authentification du proxy pour le magasin de clés externe.

- Lorsque vous connectez un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC à son proxy de magasin de clés externe, AWS KMS procède ainsi :
  - Il confirme que le domaine pour le nom DNS privé spécifié dans le point de terminaison d'URI de proxy est vérifié.
  - Il crée un point de terminaison d'interface à partir d'un VPC AWS KMS vers votre service de point de terminaison d'un VPC.
  - Il crée une zone hébergée privée pour le nom DNS privé spécifié dans le point de terminaison d'URI de proxy.
  - Envoie une GetHealthStatusdemande au proxy de stockage de clés externe. Une réponse positive du proxy confirme que le point de terminaison d'URI de proxy et le chemin d'URI de proxy sont exacts et accessibles, et que le proxy a authentifié la requête signée à l'aide des informations d'identification pour l'authentification du proxy pour le magasin de clés externe.

L'opération de connexion lance le processus de connexion de votre magasin de clés personnalisé, mais la connexion d'un magasin de clés externe à son proxy externe prend environ cinq minutes. Une réponse positive à l'opération de connexion n'indique pas que le magasin de clés externe est connecté. Pour confirmer que la connexion a été établie, utilisez la AWS KMS console ou l'DescribeCustomKeyStoresopération pour afficher l'état de connexion de votre magasin de clés externe.

Lorsque l'état de la connexion est `FAILED`, un code d'erreur de connexion s'affiche dans la console AWS KMS et est ajouté à la réponse `DescribeCustomKeyStore`. Pour obtenir de l'aide sur l'interprétation des codes d'erreur de connexion, veuillez consulter la rubrique Codes d'erreur de connexion pour les magasins de clés externes.

Déconnexion d'un magasin de clés externe

Lorsque vous déconnectez un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC de son proxy de magasin de clés externe, AWS KMS supprime son point de terminaison d'interface vers le service de point de terminaison d'un VPC et supprime l'infrastructure réseau qu'il a créée pour prendre en charge la connexion. Aucun processus équivalent n'est requis pour les magasins de clés externes disposant d'une connectivité au point de terminaison public. Cette

action n'affecte pas le service de point de terminaison d'un VPC ni aucun de ses composants de support, et elle n'affecte pas le proxy de magasin de clés externe ni aucun composant externe.

Lorsque le magasin de clés externe est déconnecté, AWS KMS n'envoie aucune requête au proxy de magasin de clés externe. L'état de connexion du magasin de clés externe est `DISCONNECTED`. Les clés KMS du magasin de clés externe déconnecté sont dans un état de clé `UNAVAILABLE` (sauf si elles sont en attente de suppression), ce qui signifie qu'elles ne peuvent pas être utilisées dans des opérations cryptographiques. Toutefois, vous pouvez toujours consulter et gérer votre magasin de clés externe et ses clés KMS existantes.

L'état déconnecté est conçu pour être temporaire et réversible. Vous pouvez reconnecter votre magasin de clés externe à tout moment. En général, aucune reconfiguration n'est nécessaire. Cependant, si des propriétés du proxy de magasin de clés externe associé ont changé pendant sa déconnexion, par exemple la rotation de ses informations d'identification pour l'authentification du proxy, vous devez modifier les paramètres du magasin de clés externe avant de le reconnecter.

> **Note**
>
> Même si un magasin de clés personnalisé est déconnecté, toutes les tentatives de création de clés KMS dans le magasin de clés personnalisé ou d'utilisation de clés KMS existantes dans les opérations de chiffrement échouent. Cette action peut empêcher les utilisateurs de stocker des données sensibles et d'y accéder.

Pour mieux estimer l'effet de la déconnexion de votre magasin de clés externe, identifiez les clés KMS du magasin de clés externe et déterminez leur utilisation antérieure.

Vous pouvez déconnecter le magasin de clés externe pour des raisons telles que les suivantes :

- Pour modifier ses propriétés. Vous pouvez modifier le nom du magasin de clés personnalisé, le chemin d'URI de proxy et les informations d'identification pour l'authentification du proxy lorsque le magasin de clés externe est connecté. Toutefois, pour modifier le type de connectivité du proxy, le point de terminaison de l'URI de proxy ou le nom du service de point de terminaison d'un VPC, vous devez d'abord déconnecter le magasin de clés externe. Pour plus de détails, consultez Modifier les propriétés du magasin de clés externe.
- Pour arrêter toute communication entre AWS KMS et le proxy de magasin de clés externe. Vous pouvez également arrêter la communication entre AWS KMS et votre proxy en désactivant votre point de terminaison ou le service de point de terminaison d'un VPC. En outre, votre proxy de magasin de clés externe ou votre logiciel de gestion de clés pourrait fournir des mécanismes

supplémentaires pour empêcher AWS KMS de communiquer avec le proxy ou pour empêcher le proxy d'accéder à votre gestionnaire de clés externe.

- Pour désactiver toutes les clés KMS du magasin de clés externe. Vous pouvez désactiver et réactiver les clés KMS dans un magasin de clés externe à l'aide de la AWS KMS console ou de l'DisableKeyopération. Ces opérations se déroulent rapidement (sous réserve d'une éventuelle cohérence), mais elles n'agissent que sur une seule clé KMS à la fois. La déconnexion du magasin de clés externe fait passer l'état de clé de toutes les clés KMS dans le magasin de clés externe à `Unavailable`, ce qui empêche leur utilisation dans les opérations cryptographiques.
- Pour réparer un échec de tentative de connexion. Si une tentative de connexion d'un magasin de clés externe échoue (l'état de connexion du magasin de clés personnalisé est `FAILED`), vous devez déconnecter le magasin de clés externe avant d'essayer de le connecter à nouveau.

État de connexion

La connexion et la déconnexion modifient l'état de connexion de votre magasin de clés personnalisé. Les valeurs d'état de connexion sont les mêmes pour les magasins de clés AWS CloudHSM et les magasins de clés externes.

Pour afficher l'état de connexion de votre magasin de clés personnalisé, utilisez l'DescribeCustomKeyStoresopération ou la AWS KMS console. L'état de la connexion apparaît dans chaque tableau de magasin de clés personnalisé, dans la section General configuration (Configuration générale) de la page détaillée de chaque magasin de clés personnalisé et dans l'onglet Cryptographic configuration (Configuration cryptographique) des clés KMS d'un magasin de clés personnalisé. Pour plus d'informations, consultez Afficher un magasin de clés AWS CloudHSM et Afficher un magasin de clés externe.

Un magasin de clés personnalisé peut avoir l'un des états de connexion suivants :

- `CONNECTED` : le magasin de clés personnalisé est connecté à son magasin de clés de sauvegarde. Vous pouvez créer et utiliser les clés KMS dans le magasin de clés personnalisé.

  Le magasin de clés de sauvegarde d'un magasin de clés AWS CloudHSM est son cluster AWS CloudHSM associé. Le magasin de clés de sauvegarde d'un magasin de clés externe est constitué du proxy de magasin de clés externe et du gestionnaire de clés externe qu'il prend en charge.

  Un état CONNECTÉ signifie que la connexion s'est établie et que le magasin de clés personnalisé n'a pas été déconnecté intentionnellement. Cela n'indique pas que la connexion fonctionne correctement. Pour plus d'informations sur l'état du AWS CloudHSM cluster associé à votre

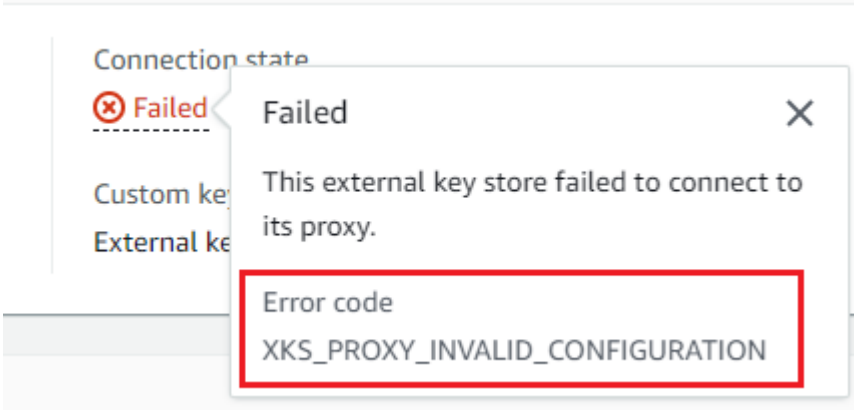
magasin de AWS CloudHSM clés, consultez la section Obtenir CloudWatch des métriques AWS CloudHSM dans le guide de AWS CloudHSM l'utilisateur. Pour plus d'informations sur l'état et le fonctionnement de votre magasin de clés externe, veuillez consulter les graphiques de la section Monitoring (Surveillance) de la page détaillée de chaque magasin de clés externe. Pour plus de détails, consultez Surveiller un magasin de clés externe.

- `CONNECTING` : le processus de connexion d'un magasin de clés personnalisé est en cours. Il s'agit d'un état transitoire.
- `DISCONNECTED`: Le magasin de clés personnalisé n'a jamais été connecté à son support, ou il a été déconnecté intentionnellement à l'aide de la AWS KMS console ou de l'DisconnectCustomKeyStoreopération.
- `DISCONNECTING` : le processus de déconnexion d'un magasin de clés personnalisé est en cours. Il s'agit d'un état transitoire.
- `FAILED` : une tentative de connexion du magasin de clés personnalisé a échoué. Le `ConnectionErrorCode` dans la DescribeCustomKeyStoresréponse indique le problème.

Pour connecter un magasin de clés personnalisé, son état de connexion doit être `DISCONNECTED`. Si l'état de la connexion est `FAILED`, utilisez le `ConnectionErrorCode` pour identifier et résoudre le problème. Déconnectez ensuite le magasin de clés personnalisé avant d'essayer de le connecter à nouveau. Pour obtenir de l'aide concernant les connexions ayant échoué, veuillez consulter Erreurs de connexion au magasin de clés externe. Pour obtenir de l'aide afin de répondre à un code d'erreur de connexion, veuillez consulter la rubrique Codes d'erreur de connexion pour les magasins de clés externes.

Pour consulter le code d'erreur de connexion, procédez comme suit :

- Dans la DescribeCustomKeyStoresréponse, visualisez la valeur de l'`ConnectionErrorCode`élément. Cet élément apparaît dans la réponse de `DescribeCustomKeyStores` uniquement lorsque le `ConnectionState` est `FAILED`.
- Pour afficher le code d'erreur de connexion dans la console AWS KMS, sur la page de détails du magasin de clés externe, passez la souris sur la valeur Failed (Échec).



Connecter un magasin de clés externe (console)

Vous pouvez utiliser la console AWS KMS pour connecter un magasin de clés externe à son proxy de magasin de clés externe.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Choisissez la ligne du magasin de clés externe que vous souhaitez connecter.

   Si l'état de connexion du magasin de clés externe est FAILED (ÉCHEC), vous devez déconnecter le magasin de clés externe avant de le connecter.
5. Dans le menu Key store actions (Actions de magasin de clés), choisissez Connect (Connecter).

Le processus de connexion prend en général environ cinq minutes. Lorsque l'opération est terminée, l'état de connexion passe à CONNECTED (CONNECTÉ).

Si l'état de connexion est Failed (Échec), passez la souris sur l'état de la connexion pour voir le code d'erreur de connexion, qui explique la cause de l'erreur. Pour obtenir de l'aide afin de répondre à un code d'erreur de connexion, veuillez consulter la rubrique Codes d'erreur de connexion pour les magasins de clés externes. Pour connecter un magasin de clés externe dont l'état de connexion est Failed (Échec), vous devez d'abord déconnecter le magasin de clés personnalisé.

Connecter un magasin de clés externe (API)

Pour connecter un magasin de clés externe déconnecté, utilisez l'ConnectCustomKeyStoreopération.

Avant la connexion, l'état de connexion du magasin de clés externe doit être `DISCONNECTED`. Si l'état actuel de la connexion est `FAILED`, déconnectez le magasin de clés externe, puis connectez-le.

Le processus de connexion prend environ cinq minutes. À moins qu'elle n'échoue rapidement, `ConnectCustomKeyStore` renvoie une réponse HTTP 200 et un objet JSON sans propriétés. Cependant, cette réponse initiale n'indique pas que la connexion a abouti. Pour déterminer si le magasin de clés externe est connecté, consultez l'état de la connexion dans la DescribeCustomKeyStoresréponse.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Pour identifier le magasin de clés externe, utilisez son ID du magasin de clés personnalisé. Vous pouvez trouver l'ID sur la page des stockages de clés personnalisés de la console ou en utilisant l'DescribeCustomKeyStoresopération. Avant d'exécuter cet exemple, remplacez l'ID de l'exemple par un ID valide.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

L'opération `ConnectCustomKeyStore` ne renvoie pas de `ConnectionState` dans sa réponse. Pour vérifier que le magasin de clés externe est connecté, utilisez l'DescribeCustomKeyStoresopération. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Toutefois, vous pouvez utiliser le paramètre `CustomKeyStoreName`ou `CustomKeyStoreId` (mais pas les deux) pour limiter la réponse à des magasins de clés personnalisés en particulier. Une valeur de `ConnectionState` égale à `CONNECTED` indique que le magasin de clés externe est connecté à son proxy de magasin de clés externe.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
```

```
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Si la valeur de `ConnectionState` dans la réponse de `DescribeCustomKeyStores` est `FAILED`, l'élément `ConnectionErrorCode` indique la raison de l'échec.

Dans l'exemple suivant, la valeur `XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND` affectée à `ConnectionErrorCode` indique que AWS KMS ne peut pas trouver le service de point de terminaison d'un VPC qu'il utilise pour communiquer avec le proxy de magasin de clés externe. Vérifiez que le `XksProxyVpcEndpointServiceName` est correct, que le principal de service AWS KMS est un principal autorisé sur le service de point de terminaison d'un Amazon VPC et que le service de point de terminaison d'un VPC ne nécessite pas l'acceptation des requêtes de connexion. Pour obtenir de l'aide afin de répondre à un code d'erreur de connexion, veuillez consulter la rubrique Codes d'erreur de connexion pour les magasins de clés externes.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
```

```
  ]
}
```

Déconnecter un magasin de clés externe (console)

Vous pouvez utiliser la console AWS KMS pour connecter un magasin de clés externe à son proxy de magasin de clés externe. Ce processus prend environ cinq minutes.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Choisissez la ligne du magasin de clés externe que vous souhaitez déconnecter.
5. Dans le menu Key store actions (Actions de magasin de clés), choisissez Disconnect (Déconnecter).

Une fois l'opération terminée, l'état de la connexion passe de DISCONNECTING à DISCONNECTED. Si l'opération échoue, un message d'erreur s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez Erreurs de connexion au magasin de clés externe.

Déconnecter un magasin de clés externe (API)

Pour déconnecter un magasin de clés externe connecté, utilisez l'DisconnectCustomKeyStoreopération. Si l'opération aboutit, AWS KMS renvoie une réponse HTTP 200 et un objet JSON sans propriétés. Le processus prend environ cinq minutes. Pour connaître l'état de connexion du magasin de clés externe, utilisez l'DescribeCustomKeyStoresopération.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Cet exemple déconnecte un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC. Avant d'exécuter cet exemple, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Pour vérifier que le magasin de clés externe est déconnecté, utilisez l'DescribeCustomKeyStoresopération. Par défaut, cette opération renvoie tous les magasins de clés personnalisés de vos compte et région. Toutefois, vous pouvez utiliser le paramètre `CustomKeyStoreName` ou `CustomKeyStoreId` (mais pas les deux) pour limiter la réponse à des magasins de clés personnalisés en particulier. La valeur de `ConnectionState` égale à `DISCONNECTED` indique que cet exemple de magasin de clés externe n'est plus connecté à son proxy de magasin de clés externe.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Supprimer un magasin de clés externe

Lorsque vous supprimez un magasin de clés externe, AWS KMS supprime toutes les métadonnées concernant le magasin de clés externe d'AWS KMS, y compris les informations sur son proxy de magasin de clés externe. Cette opération n'affecte pas le proxy du magasin de clés externe, le gestionnaire de clés externe, les clés externes ou toute ressource AWS que vous avez créée pour prendre en charge le magasin de clés externe, comme un service Amazon VPC ou un point de terminaison d'un VPC.

Avant de supprimer un magasin de clés externe, vous devez supprimer toutes les clés KMS du magasin de clés et déconnecter le magasin de clés de son proxy de magasin de clés externe. Dans le cas contraire, les tentatives de suppression du magasin de clés échouent.

La suppression d'un magasin de clés externe est irréversible, mais vous pouvez créer un autre magasin de clés externe et l'associer au même proxy de magasin de clés externe et au même gestionnaire de clés externe. Toutefois, vous ne pouvez pas recréer les clés KMS de chiffrement symétrique dans le magasin de clés externe, même si vous avez accès aux mêmes éléments de clé externe. AWS KMS inclut des métadonnées dans le texte chiffré symétrique propre à chaque clé KMS. Cette fonctionnalité de sécurité garantit que seule la clé KMS qui a chiffré des données peut les déchiffrer.

Au lieu de supprimer le magasin de clés externe, pensez à le déconnecter. Tant qu'un magasin de clés externe est déconnecté, vous pouvez gérer le magasin de clés externe et ses AWS KMS keys, mais vous ne pouvez pas créer ou utiliser des clés KMS dans le magasin de clés externe. Vous pouvez reconnecter le magasin de clés externe à tout moment et recommencer à utiliser ses clés KMS pour chiffrer et déchiffrer des données. Aucuns frais ne s'appliquent à un proxy de magasin de clés externe déconnecté ou lorsque ses clés KMS sont indisponibles.

Rubriques

- Supprimer un magasin de clés externe (console)
- Supprimer un magasin de clés externe (API)

Supprimer un magasin de clés externe (console)

Vous pouvez utiliser la console AWS KMS pour supprimer un magasin de clés externe.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Recherchez la ligne qui représente le magasin de clés externe que vous souhaitez supprimer. Si Connection state (État de connexion) du magasin de clés externe n'est pas DISCONNECTED (DÉCONNECTÉ), vous devez déconnecter le magasin de clés externe avant de le supprimer.
5. Dans le menu Key store actions (Actions de magasin de clés), choisissez Delete (Supprimer).

Une fois l'opération terminée, un message de réussite s'affiche et le magasin de clés externe n'apparaît plus dans la liste des magasins de clés. Si l'opération échoue, un message d'erreur

s'affiche qui décrit le problème et fournit une aide pour le résoudre. Si vous avez besoin d'aide supplémentaire, consultez Résoudre les problèmes liés aux magasins de clés externes.

Supprimer un magasin de clés externe (API)

Pour supprimer un magasin de clés externe, utilisez l'DeleteCustomKeyStoreopération. Si l'opération aboutit, AWS KMS renvoie une réponse HTTP 200 et un objet JSON sans propriétés.

Pour commencer, déconnectez le magasin de clés externe. Avant d'exécuter la commande, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Une fois le magasin de clés externe déconnecté, vous pouvez utiliser DeleteCustomKeyStorecette opération pour le supprimer.

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Pour confirmer que le magasin de clés externe est supprimé, utilisez l'DescribeCustomKeyStoresopération.

```
$ aws kms describe-custom-key-stores

{
    "CustomKeyStores": []
}
```

Si vous spécifiez un nom ou un ID de magasin de clés externe qui n'existe plus, AWS KMS renvoie une exception `CustomKeyStoreNotFoundException`.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0

An error occurred (CustomKeyStoreNotFoundException) when calling the
 DescribeCustomKeyStore operation:
```

Gérer des clés KMS dans un magasin de clés externe

Pour créer, afficher, gérer, utiliser et planifier la suppression des clés KMS d'un magasin de clés externe, vous devez utiliser des procédures très similaires à celles que vous utilisez pour les autres

clés KMS. Toutefois, lorsque vous créez une clé KMS dans un magasin de clés externe, vous spécifiez un magasin de clés externe et une clé externe. Lorsque vous utilisez une clé KMS dans un magasin de clés externe, les opérations de chiffrement et de déchiffrement sont effectuées par votre gestionnaire de clés externe à l'aide de la clé externe spécifiée.

AWS KMS ne peut pas créer, afficher, mettre à jour ou supprimer de clés cryptographiques dans votre gestionnaire de clés externe. AWS KMS n'accède jamais directement à votre gestionnaire de clés externe ni à aucune clé externe. Toutes les requêtes d'opérations cryptographiques sont acheminées par votre proxy de magasin de clés externe. Pour utiliser une clé KMS dans un magasin de clés externe, le magasin de clés externe qui héberge la clé KMS doit être connecté à son proxy de magasin de clés externe.

Fonctionnalités prises en charge

Outre les procédures décrites dans cette section, vous pouvez effectuer les actions suivantes avec les clés KMS d'un magasin de clés externe :

- Utiliser les politiques de clé, les politiques IAM et les octrois pour contrôler l'accès aux clés KMS.
- Activer et désactiver les clés KMS. Ces actions n'affectent pas la clé externe dans votre gestionnaire de clés externe.
- Attribuez des balises, créez des alias et utilisez le contrôle d'accès par attributs (ABAC) pour autoriser l'accès aux clés KMS.
- Utilisez les clés KMS avec les Services AWS qui s'intègrent à AWS KMS et prenez en charge les clés gérées par le client.

Fonctions non prises en charge

- Les magasins de clés externes ne prennent en charge que les clés KMS de chiffrement symétriques. Vous ne pouvez pas créer de clés KMS HMAC ou de clés KMS asymétriques dans un magasin de clés externe.
- GenerateDataKeyPairet ne GenerateDataKeyPairWithoutPlaintextsont pas pris en charge sur les clés KMS d'un magasin de clés externe.
- Vous ne pouvez pas utiliser de modèle AWS CloudFormation pour créer un magasin de clés externe ou une clé KMS dans un magasin de clés externe.
- Les clés multi-régions ne sont pas prises en charge dans un magasin de clés externe.
- Les clés KMS contenant des éléments de clé importés ne sont pas prises en charge dans un magasin de clés externe.

- La rotation automatique des clés n'est pas prise en charge pour les clés KMS dans un magasin de clés externe.

Rubriques

- Créer des clés KMS dans un magasin de clés externe
- Afficher des clés KMS dans un magasin de clés externe
- Utiliser des clés KMS dans un magasin de clés externe
- Planifier la suppression des clés KMS d'un magasin de clés externe

Créer des clés KMS dans un magasin de clés externe

Une fois que vous avez créé et connecté votre magasin de clés externe, vous pouvez créer des AWS KMS keys dans votre magasin de clés. Il doit s'agir de clés KMS de chiffrement symétrique dont la valeur d'origine est External key store (Magasin de clés externe) (`EXTERNAL_KEY_STORE`). Vous ne pouvez pas créer de clés KMS asymétriques, de clés KMS HMAC ou de clés KMS avec des éléments de clé importés dans un magasin de clé personnalisé. De plus, vous ne pouvez pas utiliser de clés KMS de chiffrement symétriques dans un magasin de clés personnalisé pour générer des paires de clés de données asymétriques.

Une clé KMS dans un magasin de clés externe peut présenter une latence, une durabilité et une disponibilité inférieures à celles d'une clé KMS standard, car elle dépend de composants situés à l'extérieur d'AWS. Avant de créer ou d'utiliser une clé KMS dans un magasin de clés externe, vérifiez que vous avez besoin d'une clé dotée de propriétés de magasin de clés externe.

> **Note**
>
> Certains gestionnaires de clés externes proposent une méthode plus simple pour créer des clés KMS dans un magasin de clés externe. Pour en savoir plus, veuillez consulter la documentation de votre gestionnaire de clés externe.

Pour créer une clé KMS dans votre magasin de clés externe, vous devez spécifier les éléments suivants :

- L'ID de votre magasin de clés externe.
- Une origine des éléments de clé du magasin de clés externe (`EXTERNAL_KEY_STORE`).

- L'ID d'une clé externe existante dans le gestionnaire de clés externe associé à votre magasin de clés externe. Cette clé externe fait office d'éléments de clé pour la clé KMS. Vous ne pouvez pas modifier l'ID de clé externe une fois que vous avez créé la clé KMS.

  AWS KMS fournit l'ID de clé externe à votre proxy de magasin de clés externe dans les requêtes d'opérations de chiffrement et de déchiffrement. AWS KMS ne peut pas accéder directement à votre gestionnaire de clés externe ni à aucune de ses clés cryptographiques.

Outre la clé externe, une clé KMS dans un magasin de clés externe contient également des éléments de clé AWS KMS. Toutes les données chiffrées au moyen de la clé KMS sont d'abord chiffrées dans AWS KMS à l'aide des éléments de clé AWS KMS de la clé, puis par votre gestionnaire de clés externe à l'aide de votre clé externe. Ce processus de double chiffrement garantit que le texte chiffré protégé par une clé KMS dans un magasin de clés externe est au moins aussi robuste que le texte chiffré protégé uniquement par AWS KMS. Pour plus de détails, consultez Fonctionnement des magasins de clés externes.

Lorsque l'opération `CreateKey` aboutit, l'état de clé de la nouvelle clé KMS est `Enabled`. Lorsque vous consultez une clé KMS dans un magasin de clés externe, vous pouvez afficher les propriétés classiques, comme son ID de clé, sa spécification de clé, son utilisation de clé, son état de clé et sa date de création. Mais vous pouvez également voir l'ID et l'état de connexion du magasin de clés externe ainsi que l'ID de la clé externe.

Si votre tentative de créer une clé KMS dans votre magasin de clés externe échoue, utilisez le message d'erreur pour identifier la cause. Il peut indiquer que le magasin de clés externe n'est pas connecté (`CustomKeyStoreInvalidStateException`), que le proxy de votre magasin de clés externe ne trouve pas de clé externe avec l'ID de clé externe spécifié (`XksKeyNotFoundException`) ou que la clé externe est déjà associée à une clé KMS dans le même magasin de clés externe `XksKeyAlreadyInUseException`.

.

Pour un exemple de journal AWS CloudTrail de l'opération qui crée une clé KMS dans un magasin de clés externe, veuillez consulter la rubrique CreateKey.

Rubriques

- Exigences relatives à une clé KMS dans un magasin de clés externe
- Créer une clé KMS dans un magasin de clés externe (console)
- Créer une clé KMS dans un magasin de clés externe (API AWS KMS)

Exigences relatives à une clé KMS dans un magasin de clés externe

Pour créer une clé KMS dans un magasin de clés externe, les propriétés suivantes sont requises pour le magasin de clés externe, la clé KMS et la clé externe qui fait office d'éléments de clé cryptographique externe pour la clé KMS.

Exigences relatives au magasin de clés externe

- Doit être connecté à son proxy de magasin de clés externe.

  Pour consulter l'état de connexion de votre magasin de clés externe, veuillez consulter la rubrique Afficher un magasin de clés externe. Pour connecter votre magasin de clés externe, veuillez consulter la rubrique Connecter et déconnecter un magasin de clés externe.

Exigences relatives aux clés KMS

Vous ne pouvez pas modifier ces propriétés après la création de la clé KMS.

- Spécification de clé : SYMMETRIC_DEFAULT
- Utilisation de clé : ENCRYPT_DECRYPT
- Origine des éléments de clé : EXTERNAL_KEY_STORE
- Multi-région : FALSE

Exigences relatives aux clés externes

- Clé cryptographique AES 256 bits (256 bits aléatoires). La propriété `KeySpec` de la clé externe doit être `AES_256`.
- Activé et disponible pour utilisation. La propriété `Status` de la clé externe doit être `ENABLED`.
- Configuré pour le chiffrement et le déchiffrement. La propriété `KeyUsage` de la clé externe doit inclure `ENCRYPT` et `DECRYPT`.
- Utilisé uniquement avec cette clé KMS. Chaque `KMS key` d'un magasin de clés externe doit être associée à une clé externe différente.

  AWS KMS recommande également que la clé externe soit utilisée exclusivement pour le magasin de clés externe. Cette restriction facilite l'identification et la résolution des problèmes liés à la clé.
- Accessible par le proxy de magasin de clés externe pour le magasin de clés externe.

Si le proxy de magasin de clés externe ne trouve pas la clé à l'aide de l'ID de clé externe spécifié, l'opération `CreateKey` échoue.

- Peut gérer le trafic anticipé généré par votre utilisation des Services AWS. AWS KMS recommande que les clés externes soient préparées à traiter jusqu'à 1 800 requêtes par seconde.

Créer une clé KMS dans un magasin de clés externe (console)

Il existe deux manières de créer une clé KMS dans un magasin de clés externe.

- Méthode 1 (recommandée) : choisissez un magasin de clés externe, puis créez une clé KMS dans ce magasin de clés externe.
- Méthode 2 : créez une clé KMS, puis indiquez qu'elle se trouve dans un magasin de clés externe.

Si vous utilisez la Méthode 1, lorsque vous choisissez votre magasin de clés externe avant de créer votre clé, AWS KMS choisit toutes les propriétés de clé KMS requises pour vous et remplit l'ID de votre magasin de clés externe. Cette méthode évite les erreurs que vous pourriez commettre lors de la création de votre clé KMS.

> **Note**
>
> N'incluez pas d'informations confidentielles ou sensibles dans l'alias, la description ou les balises. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

Méthode 1 (recommandée) : démarrer dans votre magasin de clés externe

Pour utiliser cette méthode, choisissez votre magasin de clés externe, puis créez une clé KMS. La console AWS KMS choisit toutes les propriétés requises pour vous et saisit l'ID de votre magasin de clés externe. Cette méthode évite les nombreuses erreurs que vous pourriez commettre lors de la création de votre clé KMS.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.

3. Dans le panneau de navigation, sélectionnez Custom key stores (Magasins de clés personnalisés), External key stores (Magasins de clés externes).
4. Choisissez le nom de votre magasin de clés externe.
5. Dans le coin supérieur droit, choisissez Create a KMS key in this key store (Créer une clé KMS dans ce magasin de clés).

   Si le magasin de clés externe n'est pas connecté, vous serez invité à le connecter. Si la tentative de connexion échoue, vous devez résoudre le problème et connecter le magasin de clés externe avant de pouvoir y créer une clé KMS.

   Si le magasin de clés externe est connecté, vous êtes redirigé vers la page Customer managed keys (Clés gérées par le client) pour créer une clé. Les valeurs de Key configuration (Configuration de clé) requises sont déjà choisies pour vous. En outre, l'ID du magasin de clés personnalisé de votre magasin de clés externe est renseigné, bien que vous puissiez le modifier.
6. Saisissez l'ID de clé d'une clé externe dans votre gestionnaire de clés externe. Cette clé externe doit remplir les conditions requises pour être utilisée avec une clé KMS. Vous ne pouvez pas modifier cette valeur après la création de la clé.

   Si la clé externe possède plusieurs ID, entrez l'ID de clé que le proxy de magasin de clés externe utilise pour identifier la clé externe.
7. Confirmez que vous avez l'intention de créer une clé KMS dans le magasin de clés externe spécifié.
8. Choisissez Suivant.

   Le reste de cette procédure est identique à la création d'une clé KMS standard.
9. Saisissez un alias (obligatoire) et une description (facultative) pour la clé KMS.
10. (Facultatif) Sur la page Ajouter des identifications, ajoutez des identifications qui identifient ou catégorisent votre clé KMS.

    Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour de plus amples informations sur l'étiquetage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.
11. Choisissez Suivant.

12. Dans la section Administrateurs de clé, sélectionnez les utilisateurs et les rôles IAM qui peuvent gérer la clé KMS. Pour plus d'informations, veuillez consulter la rubrique Autorise les administrateurs de clé à administrer la clé KMS.

    > **Note**
    >
    > Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation d'utiliser la clé KMS.
    > Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

13. (Facultatif) Pour empêcher les administrateurs de clé de supprimer cette clé KMS, décochez la case Allow key administrators to delete this key (Autoriser les administrateurs de clé à supprimer cette clé).

    La suppression d'une clé KMS est une opération destructrice et irréversible, qui peut rendre le texte chiffré irrécupérable. Vous ne pouvez pas recréer une clé KMS symétrique dans un magasin de clés externe, même si vous disposez des éléments de clé externe. Cependant, la suppression d'une clé KMS n'a aucun effet sur la clé externe qui lui est associée. Pour plus d'informations sur la suppression d'une clé KMS d'un magasin de clés externe, veuillez consulter la rubrique Planifier la suppression des clés KMS d'un magasin de clés externe.

14. Choisissez Suivant.

15. Dans la section Ce compte, sélectionnez les utilisateurs et rôles IAM de ce Compte AWS qui peuvent utiliser la clé KMS dans les opérations de chiffrement. Pour plus d'informations, veuillez consulter la rubrique Allows key users to use the KMS key (Autorise les utilisateurs de clé à utiliser la clé KMS).

    > **Note**
    >
    > Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation d'utiliser la clé KMS.
    > Les bonnes pratiques IAM déconseillent d'avoir recours à des utilisateurs IAM dotés d'informations d'identification à long terme. Dans la mesure du possible, utilisez des rôles IAM, qui fournissent des informations d'identification temporaires. Pour plus

d'informations, consultez Security best practices in IAM (Bonnes pratiques de sécurité dans IAM) dans le Guide de l'utilisateur IAM.

16. (Facultatif) Vous pouvez autoriser d'autres Comptes AWS à utiliser cette clé KMS pour les opérations de chiffrement. Pour cela, dans la section Autres Comptes AWS en bas de la page, sélectionnez Ajouter un autre Compte AWS et saisissez l'ID Compte AWS d'un compte externe. Pour ajouter plusieurs comptes externes, répétez cette étape.

> **Note**
>
> Les administrateurs des autres Comptes AWS doivent également autoriser l'accès à la clé KMS en créant les politiques IAM pour leurs utilisateurs. Pour plus d'informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

17. Choisissez Suivant.
18. Passez en revue les paramètres de clé que vous avez choisis. Vous pouvez toujours revenir en arrière et modifier tous les paramètres.
19. Lorsque vous avez terminé, choisissez Finish (Terminer) pour créer la clé.

Méthode 2 : démarrer avec les clés gérées par le client

Cette procédure est identique à la procédure de création d'une clé de chiffrement symétrique avec des éléments de clé AWS KMS. Mais, dans le cadre de cette procédure, vous spécifiez l'ID du magasin de clés personnalisé du magasin de clés externe et l'ID de la clé externe. Vous devez également spécifier les valeurs de propriété requises pour une clé KMS dans un magasin de clés externe, telles que la spécification de clé et l'utilisation de la clé.

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez Create key.
5. Choisissez Symmetric (Symétrique).
6. Dans Key usage (Utilisation de la clé), l'option Encrypt and decrypt (Chiffrer et déchiffrer) est sélectionnée pour vous. Ne la modifiez pas.

7. Choisissez Options avancées.
8. Pour Key material origin (Origine des éléments de clé), choisissez External key store (Magasin de clés externe).
9. Confirmez que vous avez l'intention de créer une clé KMS dans le magasin de clés externe spécifié.
10. Choisissez Suivant.
11. Choisissez la ligne qui représente le magasin de clés externe pour votre nouvelle clé KMS.

    Vous ne pouvez pas choisir un magasin de clés externe déconnecté. Pour connecter un magasin de clés déconnecté, choisissez le nom du magasin de clés, puis, dans Key store actions (Actions du magasin de clés), choisissez Connect (Connecter). Pour plus de détails, consultez Connecter un magasin de clés externe (console).
12. Saisissez l'ID de clé d'une clé externe dans votre gestionnaire de clés externe. Cette clé externe doit remplir les conditions requises pour être utilisée avec une clé KMS. Vous ne pouvez pas modifier cette valeur après la création de la clé.

    Si la clé externe possède plusieurs ID, entrez l'ID de clé que le proxy de magasin de clés externe utilise pour identifier la clé externe.
13. Choisissez Suivant.

    Le reste de cette procédure est identique à la création d'une clé KMS standard.
14. Saisissez un alias et éventuellement une description pour la clé KMS.
15. (Facultatif). Sur la page Ajouter des identifications, ajoutez des identifications qui identifient ou catégorisent votre clé KMS.

    Lorsque vous ajoutez des balises à vos ressources AWS, AWS génère un rapport de répartition des coûts faisant apparaître la consommation et les coûts regroupés par balises. Les balises peuvent également être utilisées pour contrôler l'accès à une clé KMS. Pour de plus amples informations sur l'étiquetage des clés KMS, veuillez consulter Clés de balisage et ABAC pour AWS KMS.
16. Choisissez Suivant.
17. Dans la section Administrateurs de clé, sélectionnez les utilisateurs et les rôles IAM qui peuvent gérer la clé KMS. Pour plus d'informations, veuillez consulter la rubrique Autorise les administrateurs de clé à administrer la clé KMS.

> **Note**
>
> Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation d'utiliser la clé KMS.

18. (Facultatif) Pour empêcher les administrateurs de clé de supprimer cette clé KMS, décochez la case Allow key administrators to delete this key (Autoriser les administrateurs de clé à supprimer cette clé).

    La suppression d'une clé KMS est une opération destructrice et irréversible, qui peut rendre le texte chiffré irrécupérable. Vous ne pouvez pas recréer une clé KMS symétrique dans un magasin de clés externe, même si vous disposez des éléments de clé externe. Cependant, la suppression d'une clé KMS n'a aucun effet sur la clé externe qui lui est associée. Pour plus d'informations sur la suppression d'une clé KMS d'un magasin de clés externe, veuillez consulter la rubrique Planifier la suppression des clés KMS d'un magasin de clés externe.

19. Choisissez Suivant.

20. Dans la section Ce compte, sélectionnez les utilisateurs et rôles IAM de ce Compte AWS qui peuvent utiliser la clé KMS dans les opérations de chiffrement. Pour plus d'informations, veuillez consulter la rubrique Allows key users to use the KMS key (Autorise les utilisateurs de clé à utiliser la clé KMS).

    > **Note**
    >
    > Les politiques IAM peuvent accorder à d'autres utilisateurs et rôles IAM l'autorisation d'utiliser la clé KMS.

21. (Facultatif) Vous pouvez autoriser d'autres Comptes AWS à utiliser cette clé KMS pour les opérations de chiffrement. Pour cela, dans la section Autres Comptes AWS en bas de la page, sélectionnez Ajouter un autre Compte AWS et saisissez l'ID Compte AWS d'un compte externe. Pour ajouter plusieurs comptes externes, répétez cette étape.

    > **Note**
    >
    > Les administrateurs des autres Comptes AWS doivent également autoriser l'accès à la clé KMS en créant les politiques IAM pour leurs utilisateurs. Pour plus d’informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS.

22. Choisissez Suivant.
23. Passez en revue les paramètres de clé que vous avez choisis. Vous pouvez toujours revenir en arrière et modifier tous les paramètres.
24. Lorsque vous avez terminé, choisissez Finish (Terminer) pour créer la clé.

Lorsque la procédure réussit, l'écran affiche la nouvelle clé KMS dans le magasin de clés externe que vous avez choisi. Lorsque vous choisissez le nom ou l'alias de la nouvelle clé KMS, l'onglet Cryptographic configuration (Configuration cryptographique) de sa page de détails affiche l'origine de la clé KMS (External key store [Magasin de clés externe]), le nom, l'ID et le type du magasin de clés personnalisé, et l'ID, l'utilisation de clé et l'état de la clé externe. Si la procédure échoue, un message d'erreur s'affiche qui décrit l'échec. Pour , veuillez consulter la rubrique Résoudre les problèmes liés aux magasins de clés externes.

> **Tip**
>
> Pour faciliter l'identification des clés KMS dans un magasin de clés personnalisé, sur la page Customer managed keys (Clés gérées par le client), ajoutez les colonnes Origin (Origine) et Custom key store ID (ID de magasin de clés personnalisé) à l'affichage. Pour modifier les champs du tableau, cliquez sur l'icône d'engrenage dans le coin supérieur droit de la page. Pour plus de détails, consultez Personnalisation de vos tables de clés KMS.

Créer une clé KMS dans un magasin de clés externe (API AWS KMS)

Pour créer une nouvelle clé KMS dans un magasin de clés externe, utilisez l'CreateKeyopération. Les paramètres suivants sont obligatoires :

- La valeur `Origin` doit être `EXTERNAL_KEY_STORE`.
- Le paramètre `CustomKeyStoreId` identifie votre magasin de clés externe. La valeur `ConnectionState` du magasin de clés externe spécifié doit être `CONNECTED`. Pour trouver les valeurs de `CustomKeyStoreId` et de `ConnectionState`, utilisez l'opération `DescribeCustomKeyStores`.
- Le paramètre `XksKeyId` identifie la clé externe. Cette clé externe doit remplir les conditions requises pour être associée à une clé KMS.

Vous pouvez également utiliser n'importe lequel des paramètres facultatifs de l'opération `CreateKey`, tels que les paramètres `Policy` ou Balises.

> **Note**
>
> N'incluez pas d'informations confidentielles ou sensibles dans les champs `Description` ou `Tags`. Ces champs peuvent apparaître en texte brut dans CloudTrail les journaux et autres sorties.

Les exemples de cette section utilisent la AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge.

Cet exemple de commande utilise l'CreateKeyopération pour créer une clé KMS dans un magasin de clés externe. La réponse contient les propriétés des clés KMS, l'ID du magasin de clés externe, ainsi que l'ID, l'utilisation et l'état de la clé externe. Pour obtenir des informations détaillées sur ces champs, veuillez consulter la rubrique Afficher des clés KMS dans un magasin de clés externe.

Avant d'exécuter cette commande, remplacez l'exemple d'ID de magasin de clés personnalisé par un ID valide.

```
$ aws kms create-key --origin EXTERNAL_KEY_STORE --custom-key-store-
id cks-1234567890abcdef0 --xks-key-id bb8562717f809024
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
```

```
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

Afficher des clés KMS dans un magasin de clés externe

Pour afficher les clés KMS dans un magasin de clés externe, utilisez la AWS KMS console ou l'DescribeKeyopération. Vous pouvez utiliser les mêmes techniques que celles que vous utiliseriez pour visualiser n'importe quelles clés gérées par le client AWS KMS. Consultez le Affichage des clés pour en savoir plus.

Dans la console AWS KMS, les clés KMS de votre magasin de clés externe sont affichées sur la page Customer managed keys (Clés gérées par le client), avec toutes les autres clés gérées par le client dans votre Compte AWS et votre région. Pour identifier les clés KMS dans un magasin de clés externe, filtrez par valeur d'origine distinctive, External key store (Magasin de clés externe) et ID du magasin de clés personnalisé.

Pour plus d’informations, consultez Afficher un magasin de clés externe, Surveiller un magasin de clés externe et Journalisation des appels d'API AWS KMS avec AWS CloudTrail.

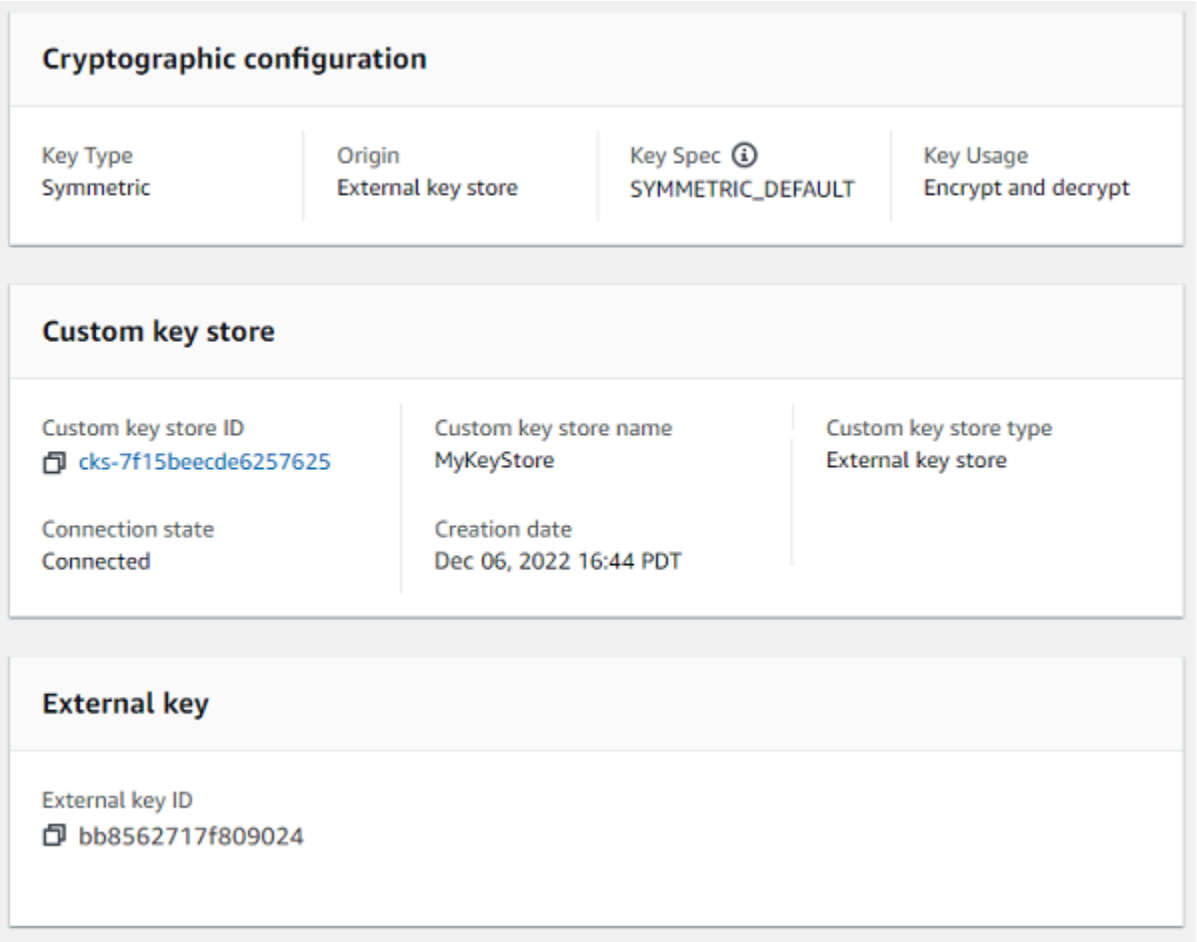
Rubriques

- Propriétés des clés KMS dans un magasin de clés externe
- Afficher des clés KMS dans un magasin de clés externe (console)
- Afficher des clés KMS dans un magasin de clés externe (API AWS KMS)

Propriétés des clés KMS dans un magasin de clés externe

Comme toutes les clés KMS, les clés KMS d'un magasin de clés externe possèdent un ARN de clé, une spécification de clé et des valeurs d'utilisation de la clé, mais elles possèdent également des propriétés et des valeurs de propriété spécifiques aux clés KMS d'un magasin de clés externe. Par exemple, la valeur Origin (Origine) de toutes les clés KMS présentes dans des magasins de clés externes est External key store (Magasin de clés externe).

Pour une clé KMS dans un magasin de clés externe, l'onglet Cryptographic configuration (Configuration cryptographique) de la console AWS KMS contient deux sections supplémentaires, Custom key store (Magasin de clés personnalisé) et External key (Clé externe).



Propriétés du magasin de clés personnalisé

Les valeurs suivantes apparaissent dans la section Stockage de clés personnalisé de l'onglet Configuration cryptographique et dans la DescribeKeyréponse. Ces propriétés s'appliquent à tous les magasins de clés personnalisés, y compris les magasins de clés AWS CloudHSM et les magasins de clés externes.

ID du magasin de clés personnalisé

Un ID unique qu'AWS KMS attribue au magasin de clés personnalisé.

Nom du magasin de clés personnalisé

Un nom convivial que vous attribuez au magasin de clés personnalisé lorsque vous le créez. Vous pouvez modifier cette valeur à tout moment.

Type de magasin de clés personnalisé

Le type de magasin de clés personnalisé. Les valeurs valides sont AWS CloudHSM (`AWS_CLOUDHSM`) ou Magasin de clés externe (`EXTERNAL_KEY_STORE`). Vous ne pouvez pas modifier le type après avoir créé le magasin de clés personnalisé.

Date de création

La date à laquelle le magasin de clés personnalisé a été créé. Cette date est affichée en heure locale pour l'Région AWS.

État de connexion

Indique si le magasin de clés personnalisé est connecté au magasin de clés de sauvegarde. L'état de connexion est `DISCONNECTED` uniquement si le magasin de clés personnalisé n'a jamais été connecté à son magasin de clés de sauvegarde, ou s'il a été déconnecté intentionnellement. Pour plus de détails, consultez the section called “État de connexion”.

Propriétés de clé externe

Les propriétés des clés externes apparaissent dans la section Clé externe de l'onglet Configuration cryptographique et dans l'`XksKeyConfiguration`élément de DescribeKeyréponse.

La section External key (Clé externe) n'apparaît dans la console AWS KMS que pour les clés KMS des magasins de clés externes. Elle fournit des informations sur la clé externe associée à la clé KMS. La clé externe est une clé cryptographique à l'extérieur d'AWS qui fait office d'éléments de clé pour la clé KMS du magasin de clés externe. Lorsque vous chiffrez ou déchiffrez à l'aide de la clé KMS, l'opération est exécutée par votre gestionnaire de clés externe à l'aide de la clé externe spécifiée.

Les valeurs suivantes apparaissent dans la section External key (Clé externe).

ID de clé externe

L'identifiant de la clé externe dans son gestionnaire de clés externe. Il s'agit de la valeur que le proxy de magasin de clés externe utilise pour identifier la clé externe. Vous choisissez l'ID de la clé externe lorsque vous créez la clé KMS et vous ne pouvez pas le modifier. Si la valeur d'ID de clé externe que vous avez utilisée pour créer la clé KMS change ou devient invalide, vous devez planifier la suppression de la clé KMS et créer une clé KMS avec la valeur d'ID de clé externe correcte.

Afficher des clés KMS dans un magasin de clés externe (console)

Afficher les clés KMS dans un magasin de clés externe (console)

1. Ouvrez la console AWS KMS à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Pour identifier les clés KMS dans votre magasin de clés externe, ajoutez les champs Origin (Origine) et Custom key store ID (ID de magasin de clés personnalisé) à votre tableau de clés. Les clés KMS dans tout magasin de clés externe ont une valeur d'Origin (Origine) égale à External key store (Magasin de clés externe).

   Dans le coin supérieur droit, choisissez l'icône d'engrenage, choisissez Origin (Origine) et Custom key store ID (ID de magasin de clés personnalisé), puis choisissez Confirm (Confirmer).
5. Choisissez l'alias ou l'ID de clé d'une clé KMS dans un magasin de clés externe.
6. Pour afficher les propriétés spécifiques aux clés KMS dans un magasin de clés externe, choisissez l'onglet Cryptographic configuration (Configuration cryptographique). Les valeurs spéciales des clés KMS d'un magasin de clés externe apparaissent dans les sections Custom key store (Magasin de clés personnalisé) et External key (Clé externe).

Afficher des clés KMS dans un magasin de clés externe (API AWS KMS)

Afficher les clés KMS dans un magasin de clés externe (API)

Vous utilisez les mêmes opérations d'AWS KMSAPI pour afficher les clés KMS dans un magasin de clés externe que vous utiliseriez pour n'importe quelle clé KMS ListKeys, y compris DescribeKey, et GetKeyPolicy. Par exemple, l'opération `describe-key` suivante de l'AWS CLI montre les champs spéciaux d'une clé KMS dans un magasin de clés externe. Avant d'exécuter une telle commande, remplacez l'exemple d'ID de clé KMS par une valeur valide.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
```

```
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```


Utiliser des clés KMS dans un magasin de clés externe

Après avoir créé une clé KMS de chiffrement symétrique dans un magasin de clés externe, vous pouvez l'utiliser pour les opérations cryptographiques suivantes :

- Encrypt
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Les opérations de chiffrement symétriques qui génèrent des paires de clés de données asymétriques GenerateDataKeyPairet GenerateDataKeyPairWithoutPlaintextne sont pas prises en charge dans les magasins de clés personnalisés.

Un contexte de chiffrement est pris en charge pour toutes les opérations cryptographiques utilisant des clés KMS dans un magasin de clés externe. Comme toujours, l'utilisation d'un contexte de chiffrement est une bonne pratique de sécurité recommandée par AWS KMS.

Lorsque vous utilisez votre clé KMS dans une requête, identifiez la clé KMS par son ID de clé, son ARN de clé, son alias ou son ARN d'alias. Vous n'avez pas besoin de spécifier le magasin de clés

externe. La réponse inclut les mêmes champs qui sont renvoyés pour une clé KMS de chiffrement symétrique. Toutefois, lorsque vous utilisez une clé KMS dans un magasin de clés externe, les opérations de chiffrement et de déchiffrement sont effectuées par votre gestionnaire de clés externe au moyen de la clé externe associée à la clé KMS.

Pour garantir que le texte chiffré au moyen d'une clé KMS dans un magasin de clés externe est au moins aussi robuste que tout texte chiffré au moyen d'une clé KMS standard, AWS KMS utilise le double chiffrement. Les données sont d'abord chiffrées dans AWS KMS en utilisant les éléments de clé AWS KMS. Elles sont ensuite chiffrées par votre gestionnaire de clés externe à l'aide de la clé externe de la clé KMS. Pour déchiffrer un texte chiffré à double chiffrement, le texte chiffré est d'abord déchiffré par votre gestionnaire de clés externe à l'aide de la clé externe de la clé KMS. Ensuite, il est déchiffré dans AWS KMS en utilisant les éléments de clé AWS KMS de la clé KMS.

Pour que cela soit possible, les conditions suivantes sont requises.

- L'état de la clé KMS doit être `Enabled`. Pour connaître l'état de la clé, consultez le champ État pour les clés gérées par le client, sur la AWS KMSconsole ou dans le `KeyState` champ de la DescribeKeyréponse.
- Le magasin de clés externe qui héberge la clé KMS doit être connecté à son proxy de magasin de clés externe, c'est-à-dire que l'état de connexion du magasin de clés externe doit être `CONNECTED`.

  Vous pouvez consulter l'état de la connexion sur la page Stockages de clés externes de la AWS KMS console ou dans la DescribeCustomKeyStoresréponse. L'état de connexion du magasin de clés externe est également affiché sur la page de détails de la clé KMS sur la console AWS KMS. Sur la page de détails, choisissez l'onglet Cryptographic configuration (Configuration cryptographique) et consultez le champ Connection state (État de la connexion) dans la section Custom key store (Magasin de clés personnalisé).

  Si l'état de connexion est `DISCONNECTED`, vous devez d'abord le connecter. Si l'état de connexion est `FAILED`, vous devez résoudre le problème, déconnecter le magasin de clés externe, puis le connecter. Pour obtenir des instructions, veuillez consulter la rubrique Connecter et déconnecter un magasin de clés externe.
- Le proxy de magasin de clés externe doit être en mesure de trouver la clé externe.
- La clé externe doit être activée et elle doit effectuer le chiffrement et le déchiffrement.

  L'état de la clé externe est indépendant et n'est pas affecté par les modifications de l'état de clé de la clé KMS, y compris par l'activation et la désactivation de la clé KMS. De même, la désactivation

ou la suppression de la clé externe ne modifie pas l'état de la clé KMS, mais les opérations cryptographiques utilisant la clé KMS associée échoueront.

Si ces conditions ne sont pas satisfaites, l'opération de chiffrement échoue, et AWS KMS renvoie une exception `KMSInvalidStateException`. Vous devrez peut-être reconnecter le magasin de clés externe ou utiliser les outils de votre gestionnaire de clés externe pour reconfigurer ou réparer votre clé externe. Pour obtenir de l'aide supplémentaire, consultez the section called “Résoudre les problèmes liés aux magasins de clés externes”.

Lorsque vous utilisez les clés KMS dans un magasin de clés externe, sachez que les clés KMS de chaque magasin de clés externe partagent un quota de magasin de clés personnalisé sur les requêtes d'opérations cryptographiques. Si vous dépassez le quota, AWS KMS renvoie un `ThrottlingException`. Pour plus d'informations sur le quota de magasin de clés personnalisé, veuillez consulter la rubrique Quotas de demandes de magasin de clés personnalisé.

Planifier la suppression des clés KMS d'un magasin de clés externe

Lorsque vous avez la certitude que vous n'aurez pas besoin d'utiliser une AWS KMS key pour les opérations de chiffrement, vous pouvez planifier la suppression de la clé KMS. Utilisez la même procédure que vous utilisez pour planifier la suppression d'une clé KMS à partir de AWS KMS. La suppression d'une clé KMS d'un magasin de clés externe n'a aucun effet sur la clé externe qui lui a servi d'élément de clé.

Vous pouvez annuler la suppression planifiée d'une clé KMS pendant sa période d'attente obligatoire. Toutefois, une clé KMS supprimée ne peut pas être récupérée. Vous ne pouvez pas recréer une clé KMS de chiffrement symétrique dans un magasin de clés externe, même si vous utilisez la même clé externe. Étant donné que chaque clé KMS symétrique d'un magasin de clés externe possède des éléments de clé AWS KMS et des métadonnées uniques, seule la clé AWS KMS qui a chiffré un texte chiffré symétrique peut le déchiffrer.

> **Warning**
>
> La suppression d'une clé KMS est une opération destructrice et potentiellement dangereuse, qui vous empêche de récupérer toutes les données chiffrées à l'aide de la clé KMS. Avant de planifier la suppression de la clé KMS, examinez l'utilisation passée de la clé KMS et créez une CloudWatch alarme Amazon qui vous avertit lorsque quelqu'un essaie d'utiliser la clé KMS alors qu'elle est en attente de suppression. Chaque fois que possible, désactivez la clé KMS, au lieu de la supprimer.

Lorsque vous planifiez la suppression d'une clé KMS d'un magasin de clés externe, son état de clé passe à Pending deletion (Suppression en attente). La clé KMS reste à l'état Pending deletion (Suppression en attente) tout au long de la période d'attente, même si la clé KMS n'est pas disponible parce que vous avez déconnecté le magasin de clés externe. Cela vous permet d'annuler la suppression de la clé KMS à tout moment au cours de la période d'attente. Lorsque la période d'attente expire, AWS KMS supprime la clé KMS depuis AWS KMS.

Lorsque vous planifiez la suppression d'une clé KMS d'un magasin de clés externe, la clé KMS devient immédiatement inutilisable (sous réserve d'une éventuelle cohérence). Toutefois, les ressources chiffrées à l'aide de clés de données protégées par la clé KMS ne sont pas affectées tant que la clé KMS n'est pas réutilisée, par exemple pour déchiffrer la clé de données. Ce problème affecte les Services AWS, dont beaucoup utilisent des clés de données pour protéger vos ressources. Pour plus de détails, consultez Comment les clés KMS inutilisables affectent les clés de données.

Vous pouvez contrôler la planification, l'annulation et la suppression de la clé KMS dans vos journaux AWS CloudTrail.

Résoudre les problèmes liés aux magasins de clés externes

La résolution de la plupart des problèmes liés aux magasins de clés externes est indiquée par le message d'erreur qu'AWS KMS affiche avec chaque exception, ou par le code d'erreur de connexion qu'AWS KMS renvoie lorsqu'une tentative de connexion du magasin de clés externe à son proxy de magasin de clés externe échoue. Toutefois, certains problèmes sont un peu plus complexes.

Lorsque vous diagnostiquez un problème lié à un magasin de clés externe, commencez par en rechercher la cause. Cela réduira le champ des possibles et rendra votre dépannage plus efficace.

- AWS KMS : le problème est peut-être interne à AWS KMS, par exemple une valeur incorrecte dans la configuration de votre magasin de clés externe.
- Externe : le problème peut provenir de l'extérieur d'AWS KMS, notamment de problèmes liés à la configuration ou au fonctionnement du proxy de magasin de clés externe, du gestionnaire de clés externe, des clés externes ou du service de point de terminaison d'un VPC.
- Réseau : il peut s'agir d'un problème de connectivité ou de mise en réseau, par exemple un problème lié à votre point de terminaison de proxy, à votre port ou à votre nom ou domaine DNS privé.

> **Note**
>
> Lorsque les opérations de gestion sur des magasins de clés externes échouent, elles génèrent plusieurs exceptions différentes. Mais les opérations de chiffrement AWS KMS renvoient une exception `KMSInvalidStateException` pour tous les échecs liés à la configuration externe ou à l'état de connexion du magasin de clés externes. Pour identifier le problème, utilisez le texte du message d'erreur qui l'accompagne. L'ConnectCustomKeyStoreopération réussit rapidement avant que le processus de connexion ne soit terminé. Pour déterminer si le processus de connexion est réussi, consultez l'état de connexion du magasin de clés externe. Si le processus de connexion échoue, AWS KMS renvoie un code d'erreur de connexion qui explique la cause et suggère une solution.

Rubriques

- Outils de dépannage pour les magasins de clés externes
- Erreurs de configuration
- Erreurs de connexion au magasin de clés externe
- Erreurs de latence et de délai d'expiration
- Erreurs liées aux informations d'identification pour l'authentification
- Erreurs d'état des clés
- Erreurs de déchiffrement
- Erreurs liées aux clés externes
- Problèmes liés aux proxys
- Problèmes d'autorisation du proxy

Outils de dépannage pour les magasins de clés externes

AWS KMS fournit plusieurs outils pour vous aider à identifier et à résoudre les problèmes liés à votre magasin de clés externe et à ses clés. Utilisez ces outils conjointement avec les outils fournis avec votre proxy de magasin de clés externe et votre gestionnaire de clés externe.

> **Note**
>
> Votre proxy de magasin de clés externe et votre gestionnaire de clés externe peuvent fournir des méthodes plus simples pour créer et gérer votre magasin de clés externe et ses clés KMS. Pour plus de détails, veuillez consulter la documentation de vos outils externes.

Exceptions AWS KMS et messages d'erreur

AWS KMS fournit un message d'erreur détaillé concernant tout problème rencontré. Vous trouverez des informations supplémentaires sur les exceptions AWS KMS dans la référence des API AWS Key Management Service et les kits SDK AWS. Même si vous utilisez la console AWS KMS, ces références peuvent vous être utiles. Par exemple, veuillez consulter la liste des erreurs correspondant à l'opération `CreateCustomKeyStores`.

Si le problème survient dans un autre service AWS, par exemple lorsque vous utilisez une clé KMS dans votre magasin de clés externe pour protéger une ressource d'un autre service AWS, le service AWS peut fournir des informations supplémentaires pour vous aider à identifier le problème. Si le AWS service ne fournit pas le message, vous pouvez consulter le message d'erreur dans les CloudTrail journaux qui enregistrent l'utilisation de votre clé KMS.

CloudTrail journaux

Chaque opération d'API AWS KMS, y compris les actions de la console AWS KMS, est enregistrée dans des journaux AWS CloudTrail. AWS KMS enregistre une entrée de journal pour les opérations réussies et échouées. Pour les opérations ayant échoué, l'entrée du journal inclut le nom de l'exception AWS KMS (`errorCode`) et le message d'erreur (`errorMessage`). Vous pouvez utiliser ces informations pour vous aider à identifier et résoudre l'erreur. Pour obtenir un exemple, consultez Échec lors du déchiffrement avec une clé KMS dans un magasin de clés externe.

L'entrée du journal inclut également l'ID de requête. Si la requête a atteint le proxy de votre magasin de clés externe, vous pouvez utiliser l'ID de requête indiqué dans l'entrée du journal pour rechercher la requête correspondante dans vos journaux de proxy, si votre proxy les fournit.

CloudWatch métriques

AWS KMSenregistre des CloudWatch statistiques Amazon détaillées concernant le fonctionnement et les performances de votre magasin de clés externe, notamment la latence, les limitations, les erreurs de proxy, le statut du gestionnaire de clés externe, le nombre de jours

avant l'expiration de votre certificat TLS et l'âge indiqué de vos informations d'authentification de proxy. Vous pouvez utiliser ces mesures pour développer des modèles de données pour le fonctionnement de votre banque de clés externe et des CloudWatch alarmes qui vous alertent des problèmes imminents avant qu'ils ne surviennent.

> **Important**
>
> AWS KMSvous recommande de créer des CloudWatch alarmes pour surveiller les métriques du magasin de clés externe. Ces alertes vous signaleront les signes précurseurs de problèmes avant qu'ils ne se produisent.

Graphiques de surveillance

AWS KMSaffiche des graphiques des CloudWatch statistiques du magasin de clés externe sur la page détaillée de chaque magasin de clés externe de la AWS KMS console. Vous pouvez utiliser les données des graphiques pour localiser la source des erreurs, détecter les problèmes imminents, établir des bases de référence et affiner vos seuils CloudWatch d'alarme. Pour plus de détails sur l'interprétation des graphiques de surveillance et l'utilisation de leurs données, veuillez consulter la rubrique Surveiller un magasin de clés externe.

Affichages des magasins de clés externes et des clés KMS

AWS KMSaffiche des informations détaillées sur vos magasins de clés externes et les clés KMS dans le magasin de clés externe de la AWS KMS console, ainsi que dans la réponse aux DescribeKeyopérations DescribeCustomKeyStoreset. Ces affichages incluent des champs spéciaux pour les magasins de clés externes et les clés KMS contenant des informations que vous pouvez utiliser pour le dépannage, telles que l'état de connexion du magasin de clés externe et l'ID de la clé externe associée à la clé KMS. Pour plus d'informations, consultez Afficher un magasin de clés externe et Afficher des clés KMS dans un magasin de clés externe.

Client de test du proxy XKS (langue française non garantie)

AWS KMS fournit un client de test open source qui vérifie que votre proxy de magasin de clés externe est conforme à la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie). Vous pouvez utiliser ce client de test pour identifier et résoudre les problèmes liés au proxy de votre magasin de clés externe.

Erreurs de configuration

Lorsque vous créez un magasin de clés externe, vous spécifiez les valeurs des propriétés qui constituent la configuration de votre magasin de clés externe, telles que les informations d'identification pour l'authentification du proxy, le point de terminaison d'URI de proxy, le chemin d'URI de proxy et le nom du service de point de terminaison d'un VPC. Lorsqu'AWS KMS détecte une erreur dans la valeur d'une propriété, l'opération échoue et renvoie une erreur qui indique la valeur défectueuse.

De nombreux problèmes de configuration peuvent être résolus en corrigeant la valeur incorrecte. Vous pouvez corriger un chemin d'URI de proxy ou des informations d'identification pour l'authentification de proxy non valide sans déconnecter le magasin de clés externe. Pour les définitions de ces valeurs, y compris les exigences d'unicité, veuillez consulter la rubrique Rassembler les conditions requises. Pour obtenir des instructions sur la mise à jour de ces valeurs, veuillez consulter la rubrique Modifier les propriétés du magasin de clés externe.

Pour éviter les erreurs liées au chemin de l'URI de votre proxy et aux valeurs des informations d'identification pour l'authentification du proxy, lorsque vous créez ou mettez à jour votre magasin de clés externe, chargez un fichier de configuration du proxy sur la console AWS KMS. Il s'agit d'un fichier JSON, contenant le chemin d'URI de proxy et les valeurs d'informations d'identification pour l'authentification du proxy, fourni par le proxy de votre magasin de clés externe ou votre gestionnaire de clés externe. Vous ne pouvez pas utiliser un fichier de configuration de proxy pour les opérations d'API AWS KMS, mais vous pouvez utiliser les valeurs du fichier pour vous aider à fournir des valeurs de paramètres pour vos requêtes d'API qui correspondent aux valeurs de votre proxy.

Erreurs de configuration générale

Exceptions : `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (opérations cryptographiques), `XksProxyInvalidConfigurationException` (opérations de gestion, à l'exception de `CreateKey`)

Codes d'erreur de connexion : `XKS_PROXY_INVALID_CONFIGURATION`, `XKS_PROXY_INVALID_TLS_CONFIGURATION`

Pour les magasins de clés externes dotés d'une connectivité à des points de terminaison publics, AWS KMS teste les valeurs des propriétés lorsque vous créez et mettez à jour le magasin de clés externe. Pour les magasins de clés externes dotés d'une connectivité au service de point de terminaison d'un VPC, AWS KMS teste les valeurs des propriétés lorsque vous connectez et mettez à jour le magasin de clés externe.

> **Note**
>
> L'opération `ConnectCustomKeyStore`, qui est asynchrone, peut réussir même si la tentative de connexion du magasin de clés externe à son proxy de magasin de clés externe échoue. Dans ce cas, il n'y a pas d'exception, mais l'état de connexion du magasin de clés externe est Échec et un code d'erreur de connexion explique le message d'erreur. Pour plus d'informations, consultez Erreurs de connexion au magasin de clés externe.

Si AWS KMS détecte une erreur dans une valeur de propriété, l'opération échoue et renvoie `XksProxyInvalidConfigurationException` avec l'un des messages d'erreur suivants.

Le proxy de magasin de clés externe a rejeté la requête en raison d'un chemin d'URI non valide. Vérifiez le chemin de l'URI de votre magasin de clés externe et mettez-le à jour si nécessaire.

- Le chemin d'URI de proxy est le chemin de base pour les requêtes AWS KMS adressées aux API du proxy. Si ce chemin est incorrect, toutes les requêtes adressées au proxy échouent. Pour afficher le chemin actuel de l'URI de proxy pour votre magasin de clés externe, utilisez la console AWS KMS ou l'opération `DescribeCustomKeyStores`. Pour trouver le chemin d'URI de proxy correct, veuillez consulter la documentation relative au proxy de votre magasin de clés externe. Pour obtenir de l'aide pour corriger la valeur du chemin d'URI de votre proxy, veuillez consulter la rubrique Modifier les propriétés du magasin de clés externe.
- Le chemin d'URI de proxy pour le proxy de votre magasin de clés externe peut changer en fonction des mises à jour apportées à votre proxy de magasin de clés externe ou à votre gestionnaire de clés externe. Pour plus d'informations sur ces modifications, veuillez consulter la documentation de votre proxy de magasin de clés externe ou de votre gestionnaire de clés externe.

`XKS_PROXY_INVALID_TLS_CONFIGURATION`
AWS KMS ne peut pas établir de connexion TLS avec le proxy de magasin de clés externe. Vérifiez la configuration TLS, y compris son certificat.

- Tous les proxys de magasin de clés externe nécessitent un certificat TLS. Le certificat TLS doit être émis par une autorité de certification publique (CA) prise en charge pour les magasins de clés externes. Pour obtenir la liste des autorités de certification prises en charge, veuillez consulter les

Autorités de certification approuvées dans la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie).

- Pour la connectivité au point de terminaison public, le nom commun (CN) du sujet figurant sur le certificat TLS doit correspondre au nom de domaine indiqué dans le point de terminaison d'URI de proxy pour le proxy de magasin de clés externe. Par exemple, si le point de terminaison public est https://myproxy.xks.example.com, le TLS, le CN sur le certificat TLS doit être `myproxy.xks.example.com` ou `*.xks.example.com`.
- Pour la connectivité au service de point de terminaison d'un VPC, le nom commun (CN) du sujet figurant sur le certificat TLS doit correspondre au nom DNS privé de votre service de point de terminaison d'un VPC. Par exemple, si le nom DNS privé est myproxy-private.xks.example.com, le CN du certificat TLS doit être `myproxy-private.xks.example.com` ou `*.xks.example.com`.
- Le certificat TLS ne peut pas avoir expiré. Pour obtenir la date d'expiration d'un certificat TLS, utilisez des outils SSL tels qu'OpenSSL. Pour surveiller la date d'expiration d'un certificat TLS associé à un magasin de clés externe, utilisez la XksProxyCertificateDaysToExpire CloudWatch métrique. Le nombre de jours avant la date d'expiration de votre certificat TLS apparaît également dans la section Monitoring (Surveillance) de la console AWS KMS.
- Si vous utilisez une connectivité au point de terminaison public, utilisez des outils de test SSL pour tester votre configuration SSL. Les erreurs de connexion TLS peuvent résulter d'un chaînage de certificats incorrect.

Erreurs de configuration de la connectivité au service de point de terminaison d'un VPC

Exceptions : `XksProxyVpcEndpointServiceNotFoundException`, `XksProxyVpcEndpointServiceInvalidConfigurationException`

Outre les problèmes de connectivité généraux, vous pouvez rencontrer les problèmes suivants lors de la création, de la connexion ou de la mise à jour d'un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC. AWS KMS teste les valeurs des propriétés d'un magasin de clés externe avec la connectivité au service de point de terminaison d'un VPC lors de la création, de la connexion et de la mise à jour du magasin de clés externe. Lorsque les opérations de gestion échouent en raison d'erreurs de configuration, elles génèrent les exceptions suivantes :

XksProxyVpcEndpointServiceNotFoundException

Ce problème peut être dû à l'une des raisons suivantes :

- Nom de service de point de terminaison d'un VPC incorrect. Vérifiez que le nom du service de point de terminaison d'un VPC pour le magasin de clés externe est correct et qu'il correspond à la valeur de point de terminaison d'URI de proxy pour le magasin de clés externe. Pour trouver le nom du service de point de terminaison VPC, utilisez la console Amazon VPC ou l'opération. DescribeVpcEndpointServices Pour trouver le nom du service de point de terminaison VPC et le point de terminaison URI du proxy d'un magasin de clés externe existant, utilisez la AWS KMS console ou l'DescribeCustomKeyStoresopération. Pour plus de détails, consultez Afficher un magasin de clés externe.
- Le service de point de terminaison d'un VPC peut se trouver dans une Région AWS différente de celle du magasin de clés externe. Vérifiez que le service de point de terminaison d'un VPC et le magasin de clés externe se trouvent dans la même région. (Le nom externe du nom de région, tel que, fait partie du nom du service de point de terminaison VPC`us-east-1`, tel que com.amazonaws.vpce.us-east-1. vpce-svc-example.) Pour obtenir la liste des exigences relatives au service de point de terminaison d'un VPC pour un magasin de clés externe, veuillez consulter la rubrique Service de point de terminaison d’un VPC. Vous ne pouvez pas déplacer un service de point de terminaison d'un VPC ou un magasin de clés externe vers une autre région. Vous pouvez toutefois créer un magasin de clés externe dans la même région que le service de point de terminaison d'un VPC. Pour plus d'informations, consultez Configurer la connectivité au service de point de terminaison d'un VPC et Créer un magasin de clés externe.
- AWS KMS n'est pas un principal autorisé pour le service de point de terminaison d'un VPC. La liste Allow principals (Principaux autorisés) pour le service de point de terminaison d'un VPC doit inclure la valeur `cks.kms.<region>.amazonaws.com`, telle que `cks.kms.eu-west-3.amazonaws.com`. Pour obtenir des instructions sur l'ajout de cette valeur, veuillez consulter la rubrique Manage permissions (Gérer les autorisations) dans le Guide AWS PrivateLink.

XksProxyVpcEndpointServiceInvalidConfigurationException

Cette erreur se produit lorsque le service de point de terminaison d'un VPC ne répond pas à l'une des exigences suivantes :

- Le VPC nécessite au moins deux sous-réseaux privés, chacun dans une zone de disponibilité différente. Pour en savoir plus sur l'ajout d'un sous-réseau à votre VPC, veuillez consulter la rubrique Créer un sous-réseau dans votre VPC dans le Guide de l'utilisateur Amazon VPC.
- Votre type de service de point de terminaison d'un VPC doit utiliser un équilibreur de charge réseau, et pas un équilibreur de charge de passerelle.
- L'acceptation ne doit pas être requise pour le service de point de terminaison d'un VPC (Acceptance required [Acceptation requise] ne doit pas être sélectionné). Si l'acceptation manuelle de chaque requête de connexion est requise, AWS KMS ne peut pas utiliser le service de point de terminaison d'un VPC pour se connecter au proxy de magasin de clés externe. Pour plus de détails, veuillez consulter la rubrique Accept or reject connection requests (Accepter ou rejeter les requêtes de connexion) dans le Guide AWS PrivateLink.
- Le service de point de terminaison d'un VPC doit avoir un nom DNS privé qui est un sous-domaine d'un domaine public. Par exemple, si le nom DNS privé est `https://myproxy-private.xks.example.com`, les domaines `xks.example.com` ou `example.com` doivent disposer d'un serveur DNS public. Pour afficher ou modifier le nom DNS privé de votre service de point de terminaison d'un VPC, veuillez consulter la rubrique Manage DNS names for VPC endpoint services (Gérer les noms DNS pour les services de point de terminaison d'un VPC) dans le Guide AWS PrivateLink.
- Le Domain verification status (Statut de vérification du domaine) du domaine de votre nom DNS privé doit être `verified`. Pour afficher et mettre à jour le statut de vérification du domaine de nom DNS privé, veuillez consulter la rubrique Vérifier votre nom de domaine DNS privé. Il peut s'écouler quelques minutes avant que le statut de vérification mis à jour n'apparaisse après que vous ayez ajouté l'enregistrement de texte requis.

  > **Note**
  >
  > Un domaine DNS privé ne peut être vérifié que s'il s'agit du sous-domaine d'un domaine public. Sinon, le statut de vérification du domaine DNS privé ne change pas, même après avoir ajouté l'enregistrement TXT requis.

- Le nom DNS privé du service de point de terminaison d'un VPC doit correspondre à la valeur de point de terminaison d'URI de proxy pour le magasin de clés externe. Pour un magasin de clés externe doté d'une connectivité au service de point de terminaison d'un VPC, le point de terminaison d'URI de proxy doit être `https://` suivi du nom DNS privé du service de point de terminaison d'un VPC. Pour consulter la valeur du point de terminaison d'URI de proxy, veuillez consulter la rubrique Afficher un magasin de clés externe. Pour modifier la valeur du point de

terminaison d'URI de proxy, veuillez consulter la rubrique Modifier les propriétés du magasin de clés externe.

Erreurs de connexion au magasin de clés externe

Le processus de connexion d'un magasin de clés externe à son proxy de magasin de clés externe prend environ cinq minutes. Sauf si elle échoue rapidement, l'opération `ConnectCustomKeyStore` renvoie une réponse HTTP 200 et un objet JSON sans propriétés. Cependant, cette réponse initiale n'indique pas que la connexion a abouti. Pour déterminer l'état de connexion du magasin de clés externe, référez-vous à son état de connexion. Si la connexion échoue, l'état de la connexion du magasin de clés externes devient `FAILED` et AWS KMS renvoie un code d'erreur de connexion qui explique la cause de l'échec.

> **Note**
>
> Si l'état de connexion d'un magasin de clés personnalisé est `FAILED`, vous devez déconnecter le magasin de clés personnalisé avant de le reconnecter. Vous ne pouvez pas connecter un magasin de clés personnalisé avec un statut de connexion `FAILED`.

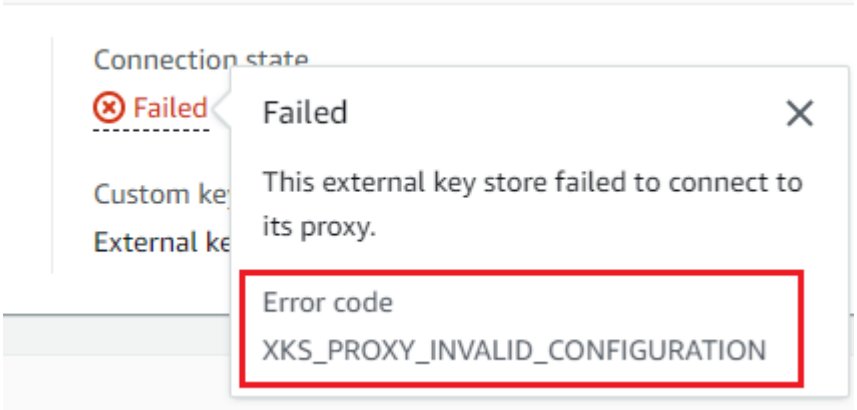
Pour consulter l'état de connexion d'un magasin de clés externe :

- Dans la DescribeCustomKeyStoresréponse, visualisez la valeur de l'`ConnectionState`élément.
- Dans la console AWS KMS, Connection state (État de la connexion) apparaît dans le tableau du magasin de clés externe. De plus, sur la page détaillée de chaque magasin de clés externe, Connection state (État de la connexion) apparaît dans la section General configuration (Configuration générale).

Lorsque l'état de connexion est `FAILED`, le code d'erreur de connexion permet d'expliquer l'erreur.

Pour consulter le code d'erreur de connexion, procédez comme suit :

- Dans la DescribeCustomKeyStoresréponse, visualisez la valeur de l'`ConnectionErrorCode`élément. Cet élément apparaît dans la réponse de `DescribeCustomKeyStores` uniquement lorsque le `ConnectionState` est `FAILED`.
- Pour afficher le code d'erreur de connexion dans la console AWS KMS, sur la page de détails du magasin de clés externe, passez la souris sur la valeur Failed (Échec).



Codes d'erreur de connexion pour les magasins de clés externes

Les codes d'erreur de connexion suivants s'appliquent aux magasins de clés externes.

`INTERNAL_ERROR`

AWS KMS n'a pas pu terminer la requête en raison d'une erreur interne. Réitérez la requête . Pour les demandes `ConnectCustomKeyStore`, déconnectez le magasin de clés personnalisé avant de tenter de vous connecter à nouveau.

`INVALID_CREDENTIALS`

L'une ou les deux valeurs de `XksProxyAuthenticationCredential` ne sont pas valides sur le proxy de magasin de clés externe spécifié.

`NETWORK_ERRORS`

Des erreurs réseau empêchent AWS KMS de connecter le magasin de clés personnalisé à son magasin de clés de sauvegarde.

`XKS_PROXY_ACCESS_DENIED`

Les requêtes AWS KMS se voient refuser l'accès au proxy de magasin de clés externe. Si le proxy de magasin de clés externe possède des règles d'autorisation, vérifiez qu'elles autorisent AWS KMS à communiquer avec le proxy en votre nom.

`XKS_PROXY_INVALID_CONFIGURATION`

Une erreur de configuration empêche le magasin de clés externe de se connecter à son proxy. Vérifiez la valeur du `XksProxyUriPath`.

`XKS_PROXY_INVALID_RESPONSE`

AWS KMS ne peut pas interpréter la réponse du proxy de magasin de clés externe. Si ce code d'erreur de connexion s'affiche à plusieurs reprises, informez-en le fournisseur du proxy de votre magasin de clés externe.

`XKS_PROXY_INVALID_TLS_CONFIGURATION`

AWS KMS ne peut pas se connecter au proxy de magasin de clés externe, car la configuration TLS n'est pas valide. Vérifiez que le proxy de magasin de clés externe prend en charge le protocole TLS 1.2 ou 1.3. Vérifiez également que le certificat TLS n'a pas expiré, qu'il correspond au nom d'hôte indiqué dans la valeur de `XksProxyUriEndpoint` et qu'il est signé par une autorité de certification approuvée figurant dans la liste des Autorités de certification approuvées (langue française non garantie).

`XKS_PROXY_NOT_REACHABLE`

AWS KMS ne peut pas communiquer avec votre proxy de magasin de clés externe. Vérifiez que les `XksProxyUriEndpoint` et `XksProxyUriPath` sont corrects. Utilisez les outils de votre proxy de magasin de clés externe pour vérifier que le proxy est actif et disponible sur son réseau. Vérifiez également que vos instances de gestionnaire de clés externe fonctionnent correctement. Les tentatives de connexion échouent avec ce code d'erreur de connexion si le proxy indique qu'aucune instance du gestionnaire de clés externe n'est disponible.

`XKS_PROXY_TIMED_OUT`

AWS KMS peut se connecter au proxy de magasin de clés externe, mais le proxy ne répond pas à AWS KMS dans le délai imparti. Si ce code d'erreur de connexion s'affiche à plusieurs reprises, informez-en le fournisseur du proxy de votre magasin de clés externe.

`XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION`

La configuration du service de point de terminaison d'un Amazon VPC n'est pas conforme aux exigences d'un magasin de clés externe AWS KMS.

- Le service de point de terminaison d'un VPC doit être un service de point de terminaison pour les points de terminaison d'interface dans l'Compte AWS de l'appelant.
- Il doit comporter un équilibreur de charge réseau (NLB) connecté à au moins deux sous-réseaux, dans deux zones de disponibilités distinctes.
- La liste `Allow principals` doit inclure le principal de service AWS KMS de la région, `cks.kms.<region>.amazonaws.com`, tel que `cks.kms.us-east-1.amazonaws.com`.

- L'acceptation des requêtes de connexion ne doit pas être requise.
- Il doit avoir un nom DNS privé. Le nom DNS privé d'un magasin de clés externe avec une connectivité `VPC_ENDPOINT_SERVICE` doit être unique dans sa Région AWS.
- La valeur de statut de vérification du domaine du nom DNS privé doit être `verified`.
- Le certificat TLS spécifie le nom d'hôte DNS privé auquel le point de terminaison est accessible.

`XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND`

AWS KMS ne trouve pas le service de point de terminaison d'un VPC qu'il utilise pour communiquer avec le proxy de magasin de clés externe. Vérifiez que le `XksProxyVpcEndpointServiceName` est correct et que le principal du service AWS KMS dispose des autorisations de consommateur de service sur le service de point de terminaison d'un Amazon VPC.

Erreurs de latence et de délai d'expiration

Exceptions : `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (opérations cryptographiques), `XksProxyUriUnreachableException` (opérations de gestion)

Codes d'erreur de connexion : `XKS_PROXY_NOT_REACHABLE`, `XKS_PROXY_TIMED_OUT`

Lorsque AWS KMS ne parvient pas à contacter le proxy dans le délai des 250 millisecondes, il renvoie une exception. Les opérations `CreateCustomKeyStore` et `UpdateCustomKeyStore` renvoient l'exception `XksProxyUriUnreachableException`. Les opérations de chiffrement renvoient l'exception standard `KMSInvalidStateException` avec un message d'erreur décrivant le problème. Si `ConnectCustomKeyStore` échoue, AWS KMS renvoie un code d'erreur de connexion décrivant le problème.

Les erreurs de délai d'expiration peuvent être des problèmes transitoires pouvant être résolus en réitérant la requête. Si le problème persiste, vérifiez que le proxy de votre magasin de clés externe est actif et connecté au réseau, et que le point de terminaison d'URI de proxy, le chemin d'URI de proxy et le nom du service de point de terminaison d'un VPC (le cas échéant) sont corrects dans votre magasin de clés externe. Vérifiez également que votre gestionnaire de clés externe est proche de l'Région AWS de votre magasin de clés externe. Si vous devez mettre à jour l'une de ces valeurs, veuillez consulter la rubrique Modifier les propriétés du magasin de clés externe.

Pour suivre les modèles de latence, utilisez la `XksProxyLatency` CloudWatch métrique et le graphique de latence moyenne (basé sur cette métrique) dans la section Surveillance de la AWS

KMS console. Votre proxy de magasin de clés externe peut également générer des journaux et des métriques permettant de suivre la latence et les délais d'expiration.

`XksProxyUriUnreachableException`
AWS KMS ne peut pas communiquer avec le proxy de magasin de clés externe. Il peut s'agir d'un problème réseau transitoire. Si cette erreur s'affiche à plusieurs reprises, vérifiez que le proxy de votre magasin de clés externe est actif et connecté au réseau, et que l'URI de son point de terminaison est correcte dans votre magasin de clés externe.

- Le proxy du magasin de clés externe n'a pas répondu à une requête API du proxy AWS KMS dans l'intervalle d'expiration de 250 millisecondes. Cela peut indiquer un problème réseau transitoire ou un problème de fonctionnement ou de performance avec le proxy. Si une nouvelle tentative ne résout pas le problème, prévenez l'administrateur de votre proxy de magasin de clés externe.

Les erreurs de latence et de délai d'expiration se manifestent souvent par des échecs de connexion. Lorsque l'ConnectCustomKeyStoreopération échoue, l'état de connexion du magasin de clés externe AWS KMS devient `FAILED` et renvoie un code d'erreur de connexion expliquant l'erreur. Pour obtenir la liste des codes d'erreur de connexion et des suggestions pour les résoudre, veuillez consulter la rubrique Codes d'erreur de connexion pour les magasins de clés externes. Les listes de codes de connexion pour All custom key stores (Tous les magasins de clés personnalisés) et les External key stores (Magasins de clés externes) s'appliquent aux magasins de clés externes. Les erreurs de connexion suivantes sont liées à la latence et aux délais d'expiration.

`XKS_PROXY_NOT_REACHABLE`

-ou-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
AWS KMS ne peut pas communiquer avec le proxy de magasin de clés externe. Vérifiez que le proxy de votre magasin de clés externe est actif et connecté au réseau, et que son chemin d'URI et son URI de point de terminaison ou son nom de service VPC sont corrects dans votre magasin de clés externe.

Cette erreur peut se produire dans les conditions suivantes :

- Le proxy de magasin de clés externe n'est pas actif et/ou n'est pas connecté au réseau.
- Une erreur s'est produite dans les valeurs du point de terminaison d'URI de proxy, du chemin d'URI de proxy ou du nom du service de point de terminaison d'un VPC (le cas échéant) dans la configuration du magasin de clés externe. Pour consulter la configuration du magasin de clés externe, utilisez l'DescribeCustomKeyStoresopération ou consultez la page détaillée du magasin de clés externe dans la AWS KMS console.
- Il se peut qu'une erreur de configuration réseau, telle qu'une erreur de port, se produise sur le chemin réseau entre AWS KMS et le proxy de magasin de clés externe. AWS KMS communique avec le proxy de magasin de clés externe sur le port 443. Cette valeur n'est pas configurable.
- Lorsque le proxy de stockage de clés externe indique (dans une GetHealthStatusréponse) que toutes les instances du gestionnaire de clés externe le sontUNAVAILABLE, l'ConnectCustomKeyStoreopération échoue avec un `ConnectionErrorCode` de`XKS_PROXY_NOT_REACHABLE`. Pour obtenir de l'aide, veuillez consulter la documentation de votre gestionnaire de clés externe.
- Cette erreur peut résulter d'une longue distance physique entre le gestionnaire de clés externe et l'Région AWS dans laquelle se trouve le magasin de clés externe. La latence du ping, c'est-à-dire le temps d'aller-retour réseau (RTT), entre l'Région AWS et le gestionnaire de clés externe ne doit pas dépasser 35 millisecondes. Vous devrez peut-être créer un magasin de clés externe dans une Région AWS plus proche du gestionnaire de clés externe, ou déplacer le gestionnaire de clés externe vers un centre de données plus proche de l'Région AWS.

`XKS_PROXY_TIMED_OUT`

-ou-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
AWS KMS a rejeté la requête, car le proxy de magasin de clés externe n'a pas répondu dans les temps. Réitérez la requête . Si cette erreur s'affiche à plusieurs reprises, signalez-la à l'adminis trateur de votre proxy de magasin de clés externe.

Cette erreur peut se produire dans les conditions suivantes :

- Cette erreur peut résulter d'une longue distance physique entre le gestionnaire de clés externe et le proxy du magasin de clés externe. Si possible, rapprochez le proxy du magasin de clés externe du gestionnaire de clés externe.
- Des erreurs de délai d'expiration peuvent survenir lorsque le proxy n'est pas conçu pour gérer le volume et la fréquence des requêtes émanant de AWS KMS. Si vos CloudWatch statistiques indiquent un problème persistant, informez-en l'administrateur proxy de votre magasin de clés externe.
- Des erreurs de délai d'expiration peuvent survenir lorsque la connexion entre le gestionnaire de clés externe et l'Amazon VPC pour le magasin de clés externe ne fonctionne pas correctement. Si vous utilisez AWS Direct Connect, vérifiez que votre VPC et votre gestionnaire de clés externe peuvent communiquer efficacement. Si vous avez besoin d'aide pour résoudre un problème, consultez Troubleshooting AWS Direct Connect (Résolution des problèmes) dans le Guide de l'utilisateur AWS Direct Connect.

`XKS_PROXY_TIMED_OUT`

-ou-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
Le proxy de magasin de clés externe n'a pas répondu à la requête dans le délai imparti. Réitérez la requête . Si cette erreur s'affiche à plusieurs reprises, signalez-la à l'administrateur de votre proxy de magasin de clés externe.

- Cette erreur peut résulter d'une longue distance physique entre le gestionnaire de clés externe et le proxy du magasin de clés externe. Si possible, rapprochez le proxy du magasin de clés externe du gestionnaire de clés externe.

Erreurs liées aux informations d'identification pour l'authentification

Exceptions : `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (opérations cryptographiques), `XksProxyIncorrectAuthenticationCredentialException` (opérations de gestion autres que `CreateKey`)

Vous établissez et gérez les informations d'identification pour l'authentification d'AWS KMS sur votre proxy de magasin de clés externe. Vous indiquez ensuite à AWS KMS les valeurs des informations d'identification lorsque vous créez un magasin de clés externe. Pour modifier les informations d'identification pour l'authentification, effectuez la modification sur votre proxy de magasin de clés externe. Ensuite, mettez à jour les informations d'identification de votre magasin de clés externe. Si votre proxy effectue une rotation des informations d'identification, vous devez mettre à jour les informations d'identification de votre magasin de clés externe.

Si le proxy de magasin de clés externe n'authentifie pas une requête signée avec les informations d'identification pour l'authentification du proxy de votre magasin de clés externe, le résultat dépend de la requête :

- `CreateCustomKeyStore` et `UpdateCustomKeyStore` échouent avec une exception `XksProxyIncorrectAuthenticationCredentialException`.
- `ConnectCustomKeyStore` réussit, mais la connexion échoue. L'état de connexion est `FAILED` et le code d'erreur de connexion est `INVALID_CREDENTIALS`. Pour plus de détails, consultez Erreurs de connexion au magasin de clés externe.
- Les opérations de chiffrement renvoient l'exception `KMSInvalidStateException` pour toutes les erreurs de configuration externe et d'état de connexion qui se produisent dans un magasin de clés externes. Le message d'erreur qui l'accompagne décrit le problème.

Le proxy de magasin de clés externe a rejeté la requête, car il n'a pas pu authentifier AWS KMS. Vérifiez les informations d'identification de votre magasin de clés externe et mettez-les à jour si nécessaire.

Cette erreur peut se produire dans les conditions suivantes :

- L'ID de clé d'accès ou la clé d'accès secrète du magasin de clés externe ne correspond pas aux valeurs établies sur le proxy de magasin de clés externe.

  Pour corriger cette erreur, mettez à jour les informations d'identification pour l'authentification du proxy de votre magasin de clés externe. Vous pouvez effectuer cette modification sans déconnecter votre magasin de clés externe.
- Un proxy inverse entre AWS KMS et le proxy de magasin de clés externe peut manipuler les en-têtes HTTP de manière à invalider les signatures SigV4. Pour corriger cette erreur, contactez l'administrateur de votre proxy.

Erreurs d'état des clés

Exceptions : `KMSInvalidStateException`

`KMSInvalidStateException` est utilisée à deux fins distinctes pour les clés KMS dans les magasins de clés personnalisés.

- Lorsqu'une opération de gestion, telle que `CancelKeyDeletion`, échoue et renvoie cette exception, cela indique que l'état de clé de la clé KMS n'est pas compatible avec l'opération.
- Lorsqu'une opération cryptographique sur une clé KMS dans un magasin de clés personnalisé échoue avec l'exception `KMSInvalidStateException`, cela peut indiquer un problème lié à l'état de la clé KMS. Cependant, l'opération de chiffrement AWS KMS renvoie l'exception `KMSInvalidStateException` pour toutes les erreurs de configuration externe et d'état de connexion qui se produisent dans un magasin de clés externes. Pour identifier le problème, utilisez le message d'erreur qui accompagne l'exception.

Pour trouver l'état de clé requis pour les opérations d'une API AWS KMS, veuillez consulter la rubrique Principaux états des clés AWS KMS. Pour obtenir l'état de clé d'une clé KMS, sur la page Clés gérées par le client, veuillez consulter le champ Status (État) de la clé KMS. Vous pouvez également utiliser l'DescribeKeyopération et afficher l'`KeyState`élément dans la réponse. Pour plus de détails, consultez Affichage des clés.

> **Note**
>
> L'état d'une clé KMS dans un magasin de clés externe n'indique rien quant au statut de la clé externe associée. Pour plus d'informations sur le statut de la clé externe, utilisez votre gestionnaire de clés externe et les outils de proxy de votre magasin de clés externe. L'exception `CustomKeyStoreInvalidStateException` fait référence à l'état de connexion du magasin de clés externe, et non à l'état de clé d'une clé KMS.

Une opération de chiffrement sur une clé KMS figurant dans un magasin personnalisé peut échouer si la clé KMS affiche l'état `Unavailable` ou `PendingDeletion`. (Les touches désactivées renvoient l'exception `DisabledException`).

- Une clé KMS possède un état `Disabled` clé uniquement lorsque vous la désactivez intentionnellement dans la AWS KMS console ou en utilisant l'DisableKeyopération. Lorsqu'une clé KMS est désactivée, vous pouvez afficher et gérer la clé, mais vous ne pouvez pas l'utiliser

dans les opérations cryptographiques. Pour résoudre ce problème, activez la clé. Pour plus de détails, consultez Activation et désactivation des clés.

- L'état de clé d'une clé KMS est `Unavailable` lorsque le magasin de clés externe est déconnecté de son proxy de magasin de clés externe. Pour corriger une clé KMS indisponible, reconnectez le magasin de clés externe. Une fois le magasin de clés externe reconnecté, l'état de clé des clés KMS du magasin de clés externe est automatiquement restauré à son état précédent, soit `Enabled` ou `Disabled`.

  L'état de clé d'une clé KMS est `PendingDeletion` lorsque sa suppression a été planifiée et qu'elle se trouve dans sa période d'attente. Une erreur d'état de clé sur une clé KMS en attente de suppression indique que la clé ne doit pas être supprimée, soit parce qu'elle est utilisée pour le chiffrement, soit parce qu'elle est requise pour le déchiffrement. Pour réactiver la clé KMS, annulez la suppression planifiée, puis activez la clé. Pour plus de détails, consultez Planification et annulation d'une suppression de clé.

Erreurs de déchiffrement

Exceptions : `KMSInvalidStateException`

Lorsqu'une opération Decrypt (Déchiffrer) avec une clé KMS figurant dans un magasin de clés externes échoue, AWS KMS renvoie l'exception standard `KMSInvalidStateException`, que les opérations de chiffrement utilisent pour toutes les erreurs de configuration externe et d'état de connexion qui se produisent dans un magasin de clés externes. Le message d'erreur signale le problème.

Pour déchiffrer un texte chiffré à l'aide d'un double chiffrement, le gestionnaire de clés externes utilise d'abord la clé externe pour déchiffrer la couche externe du texte chiffré. AWS KMS utilise ensuite les éléments de clé AWS KMS de la clé KMS pour déchiffrer la couche interne du texte chiffré. Un texte chiffré non valide ou endommagé peut être rejeté par le gestionnaire de clés externe ou par AWS KMS.

Les messages d'erreur suivants accompagnent l'exception `KMSInvalidStateException` en cas d'échec du déchiffrement. Cela indique un problème lié au texte chiffré ou au contexte de chiffrement facultatif de la requête.

Le proxy de magasin de clés externe a rejeté la requête, car le texte chiffré spécifié ou les données authentifiées supplémentaires sont endommagés, manquants ou non valides.

- Lorsque le proxy de magasin de clés externe ou le gestionnaire de clés externe signale qu'un texte chiffré ou son contexte de chiffrement n'est pas valide, cela indique généralement un problème lié au texte chiffré ou au contexte de chiffrement dans la requête `Decrypt` envoyée à AWS KMS. Pour les opérations `Decrypt`, AWS KMS envoie au proxy le même texte chiffré et le même contexte de chiffrement que ceux qu'il reçoit dans la requête `Decrypt`.

  Cette erreur peut être causée par un problème de mise en réseau lors du transit, comme une inversion de bit. Réitérez la requête `Decrypt`. Si le problème persiste, vérifiez que le texte chiffré n'a pas été altéré ou corrompu. Vérifiez également que le contexte de chiffrement dans la requête `Decrypt` adressée à AWS KMS correspond au contexte de chiffrement dans la requête qui a chiffré les données.

Le texte chiffré que le proxy de magasin de clés externe a soumis pour déchiffrement, ou le contexte de chiffrement, est endommagé, manquant ou non valide.

- Lorsqu'AWS KMS rejette le texte chiffré qu'il a reçu du proxy, cela indique que le gestionnaire de clés externe ou le proxy a renvoyé un texte chiffré non valide ou endommagé à AWS KMS.

  Cette erreur peut être causée par un problème de mise en réseau lors du transit, comme une inversion de bit. Réitérez la requête `Decrypt`. Si le problème persiste, vérifiez que le gestionnaire de clés externe fonctionne correctement et que le proxy de magasin de clés externe n'altère pas le texte chiffré qu'il reçoit du gestionnaire de clés externe avant de le renvoyer à AWS KMS.

Erreurs liées aux clés externes

Une clé externe est une clé cryptographique du gestionnaire de clés externe qui fait office d'éléments de clé externes pour une clé KMS. AWS KMS ne peut pas accéder directement à la clé externe. Il doit demander au gestionnaire de clés externe (via le proxy de magasin de clés externe) d'utiliser la clé externe pour chiffrer des données ou déchiffrer un texte chiffré.

Vous spécifiez l'ID de la clé externe dans son gestionnaire de clés externe lorsque vous créez une clé KMS dans votre magasin de clés externe. Vous ne pouvez pas modifier l'ID de la clé externe après la création de la clé KMS. Pour éviter tout problème lié à la clé KMS, l'opération `CreateKey` demande au proxy de magasin de clés externe de vérifier l'ID et la configuration de la clé externe. Si la clé externe ne répond pas aux exigences requises pour être utilisée avec une clé KMS, l'opération `CreateKey` échoue avec une exception et un message d'erreur identifiant le problème.

Cependant, des problèmes peuvent survenir après la création de la clé KMS. Si une opération de chiffrement échoue en raison d'un problème lié à la clé externe, elle renvoie une exception `KMSInvalidStateException` avec un message d'erreur indiquant le problème.

CreateKey erreurs pour la clé externe

Exceptions : `XksKeyAlreadyInUseException`, `XksKeyNotFoundException`, `XksKeyInvalidConfigurationException`

L'CreateKeyopération tente de vérifier l'ID et les propriétés de la clé externe que vous fournissez dans le paramètre ID de clé externe (console) ou `XksKeyId` (API). Cette pratique a pour but de détecter les erreurs à un stade précoce avant que vous n'essayiez d'utiliser la clé externe avec la clé KMS.

Clé externe en cours d'utilisation

Chaque clé KMS d'un magasin de clés externe doit utiliser une clé externe différente. Lorsqu'il `CreateKey` reconnaît que l'ID de clé externe (XksKeyId) d'une clé KMS n'est pas unique dans le magasin de clés externe, il échoue avec un`XksKeyAlreadyInUseException`.

Si vous utilisez plusieurs ID pour la même clé externe, `CreateKey` ne reconnaîtra pas le doublon. Cependant, les clés KMS avec la même clé externe ne sont pas interopérables, car elles ont des éléments de clé AWS KMS et des métadonnées différents.

Clé externe introuvable

Lorsque le proxy de stockage de clés externe indique qu'il ne peut pas trouver la clé externe à l'aide de l'ID de clé externe (XksKeyId) pour la clé KMS, l'`CreateKey`opération échoue et renvoie `XksKeyNotFoundException` le message d'erreur suivant.

> Le proxy de magasin de clés externe a rejeté la requête, car il n'a pas trouvé la clé externe.

Cette erreur peut se produire dans les conditions suivantes :

- L'ID de la clé externe (`XksKeyId`) de la clé KMS n'est peut-être pas valide. Pour trouver l'ID que votre proxy de clé externe utilise pour identifier la clé externe, veuillez consulter la documentation de votre proxy de magasin de clés externe ou de votre gestionnaire de clés externe.
- La clé externe peut avoir été supprimée de votre gestionnaire de clés externe. Pour effectuer des recherches, utilisez vos outils de gestionnaire de clés externe. Si la clé externe est supprimée

définitivement, utilisez une autre clé externe avec la clé KMS. Pour obtenir la liste des exigences relatives à la clé externe, veuillez consulter la rubrique Exigences relatives à une clé KMS dans un magasin de clés externe.

Exigences relatives aux clés externes non satisfaites

Lorsque le proxy du magasin de clés externes indique que la clé externe ne répond pas aux exigences définies pour une utilisation avec une clé KMS, l'opération `CreateKey` échoue et renvoie l'exception `XksKeyInvalidConfigurationException` avec l'un des messages d'erreur suivants.

La spécification de clé de la clé externe doit être AES_256. La spécification de clé de la clé externe spécifiée est *<key-spec>* .

- La clé externe doit être une clé de chiffrement symétrique de 256 bits avec une spécification de clé AES_256. Si la clé externe spécifiée est d'un type différent, spécifiez l'ID d'une clé externe qui répond à cette exigence.

Le statut de la clé externe doit être ACTIVÉ. Le statut de la clé externe spécifiée est*<status>*.

- La clé externe doit être activée dans le gestionnaire de clés externe. Si la clé externe spécifiée n'est pas activée, utilisez vos outils de gestionnaire de clés externe pour l'activer ou spécifiez une clé externe activée.

L'utilisation de la clé externe doit inclure ENCRYPT et DECRYPT. L'utilisation de la clé externe spécifiée est <*key-usage* >.

- La clé externe doit être configurée pour le chiffrement et le déchiffrement dans le gestionnaire de clés externe. Si la clé externe spécifiée n'inclut pas ces opérations, utilisez vos outils de gestionnaire de clés externe pour modifier les opérations ou spécifiez une autre clé externe.

Erreurs d'opérations cryptographiques pour la clé externe

Exceptions : `KMSInvalidStateException`

Lorsque le proxy de magasin de clés externe ne trouve pas la clé externe associée à la clé KMS, ou que la clé externe ne répond pas aux exigences requises pour être utilisée avec une clé KMS, l'opération cryptographique échoue.

Les problèmes de clé externe détectés lors d'une opération cryptographique sont plus difficiles à résoudre que les problèmes de clé externe détectés avant la création de la clé KMS. Vous ne pouvez pas modifier l'ID de la clé externe après la création de la clé KMS. Si la clé KMS n'a encore chiffré aucune donnée, vous pouvez supprimer la clé KMS et en créer une nouvelle avec un ID de clé externe différent. Cependant, le texte chiffré généré à l'aide de la clé KMS ne peut être déchiffré par aucune autre clé KMS, même si elle possède la même clé externe, car les clés auront des métadonnées et des éléments de clé AWS KMS différents. Dans la mesure du possible, utilisez plutôt vos outils de gestionnaire de clés externe pour résoudre le problème lié à la clé externe.

Lorsque le proxy du magasin de clés externes signale un problème lié à la clé externe, les opérations de chiffrement renvoient l'exception `KMSInvalidStateException` avec un message d'erreur identifiant le problème.

Clé externe introuvable

Lorsque le proxy de stockage de clés externe indique qu'il ne peut pas trouver la clé externe à l'aide de l'ID de clé externe (XksKeyId) de la clé KMS, les opérations cryptographiques renvoient un `KMSInvalidStateException` avec le message d'erreur suivant.

```
Le proxy de magasin de clés externe a rejeté la requête, car il n'a pas trouvé la clé externe.
```

Cette erreur peut se produire dans les conditions suivantes :

- L'ID de la clé externe (`XksKeyId`) de la clé KMS n'est plus valide.

  Pour trouver l'ID de clé externe associé à votre clé KMS, consultez les détails de la clé KMS. Pour trouver l'ID que votre proxy de clé externe utilise pour identifier la clé externe, veuillez consulter la documentation de votre proxy de magasin de clés externe ou de votre gestionnaire de clés externe.

  AWS KMS vérifie l'ID de clé externe lorsqu'il crée une clé KMS dans un magasin de clés externe. Cependant, l'ID peut devenir invalide, en particulier si la valeur de l'ID de clé externe est un alias ou un nom mutable. Vous ne pouvez pas modifier l'ID de clé externe associé à une clé KMS existante. Pour déchiffrer tout texte chiffré au moyen de la clé KMS, vous devez réassocier la clé externe à l'ID de la clé externe existante.

Si vous n'avez pas encore utilisé la clé KMS pour chiffrer des données, vous pouvez créer une clé KMS avec un ID de clé externe valide. Toutefois, si vous avez généré du texte chiffré à l'aide de la clé KMS, vous ne pouvez utiliser aucune autre clé KMS pour le déchiffrer, même si vous utilisez la même clé externe.

- La clé externe peut avoir été supprimée de votre gestionnaire de clés externe. Pour effectuer des recherches, utilisez vos outils de gestionnaire de clés externe. Si possible, essayez de récupérer les éléments de clé à partir d'une copie ou d'une sauvegarde de votre gestionnaire de clés externe. Si la clé externe est supprimée définitivement, tout texte chiffré au moyen de la clé KMS associée devient irrécupérable.

Erreurs de configuration des clés externes

Lorsque le proxy du magasin de clés externes indique que la clé externe ne répond pas aux exigences définies pour une utilisation avec une clé KMS, l'opération de chiffrement renvoie l'exception `KMSInvalidStateException` avec l'un des messages d'erreur suivants.

Le proxy de magasin de clés externe a rejeté la requête, car la clé externe ne prend pas en charge l'opération demandée.

- La clé externe doit prendre en charge à la fois le chiffrement et le déchiffrement. Si l'utilisation de la clé n'inclut pas le chiffrement et le déchiffrement, utilisez vos outils de gestionnaire de clés externe pour modifier l'utilisation de la clé.

Le proxy de magasin de clés externe a rejeté la requête, car la clé externe n'est pas activée dans le gestionnaire de clés externe.

- La clé externe doit être activée et disponible pour son utilisation dans le gestionnaire de clés externe. Si l'état de la clé externe n'est pas `Enabled`, utilisez vos outils de gestionnaire de clés externe pour l'activer.

Problèmes liés aux proxys

Exceptions :

`CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (opérations cryptographiques), `UnsupportedOperationException`, `XksProxyUriUnreachableException`, `XksProxyInvalidResponseException` (opérations de gestion autres que `CreateKey`)

Le proxy de magasin de clés externe assure la médiation de toutes les communications entre AWS KMS et le gestionnaire de clés externe. Il traduit également les requêtes AWS KMS génériques dans un format que votre gestionnaire de clés externe peut comprendre. Si le proxy de magasin de clés externe n'est pas conforme à la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie), s'il ne fonctionne pas correctement ou s'il ne peut pas communiquer avec AWS KMS, vous ne pourrez ni créer ni utiliser de clés KMS dans votre magasin de clés externe.

Bien que de nombreuses erreurs mentionnent le proxy de magasin de clés externe en raison de son rôle critique dans l'architecture du magasin de clés externe, ces problèmes peuvent provenir du gestionnaire de clés externe ou de la clé externe.

Les problèmes abordés dans cette section concernent des problèmes liés à la conception ou au fonctionnement du proxy de magasin de clés externe. La résolution de ces problèmes peut nécessiter une modification du logiciel de proxy. Consultez l'administrateur de votre proxy. Pour vous aider à diagnostiquer les problèmes de proxy, AWS KMS fournit XKS Proxy Text Client, un client de test open source qui vérifie si votre proxy de magasin de clés externe respecte la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie).

`CustomKeyStoreInvalidStateException`, `KMSInvalidStateException` ou `XksProxyUriUnreachableException`
Le proxy de magasin de clés externe est dans un état défectueux. Si ce message s'affiche à plusieurs reprises, prévenez l'administrateur de votre proxy de magasin de clés externe.

- Cette erreur peut indiquer un problème de fonctionnement ou une erreur logicielle dans le proxy de magasin de clés externe. Vous pouvez trouver les entrées du CloudTrail journal correspondant à l'opération d'AWS KMSAPI qui a généré chaque erreur. Cette erreur peut être résolue en réitérant l'opération. Toutefois, si le problème persiste, prévenez l'administrateur de votre proxy de magasin de clés externe.
- Lorsque le proxy de banque de clés externe indique (dans une GetHealthStatusréponse) que toutes les instances du gestionnaire de clés externe le sont`UNAVAILABLE`, les tentatives de création ou de mise à jour d'une banque de clés externe échouent, à cette exception près. Si cette erreur persiste, consultez la documentation de votre gestionnaire de clés externe.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ou `XksProxyInvalidResponseException`
AWS KMS ne peut pas interpréter la réponse du proxy de magasin de clés externe. Si cette erreur s'affiche à plusieurs reprises, contactez l'administrateur de votre proxy de magasin de clés externe.

- Les opérations AWS KMS génèrent cette exception lorsque le proxy renvoie une réponse non définie qu'AWS KMS ne peut ni analyser ni interpréter. Cette erreur peut survenir occasionnellement en raison de problèmes externes temporaires ou d'erreurs réseau sporadiques. Toutefois, s'il persiste, cela peut indiquer que le proxy de magasin de clés externe ne respecte pas la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie). Informez l'administrateur ou le fournisseur de votre magasin de clés externe.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ou `UnsupportedOperationException`

Le proxy de magasin de clés externe a rejeté la requête, car il ne prend pas en charge l'opération cryptographique demandée.

- Le proxy de magasin de clés externe doit prendre en charge toutes les API de proxy définies dans la spécification de l'API du proxy de magasin de clés externe AWS KMS (langue française non garantie). Cette erreur indique que le proxy ne prend pas en charge l'opération liée à la requête. Informez l'administrateur ou le fournisseur de votre magasin de clés externe.

Problèmes d'autorisation du proxy

Exceptions : `CustomKeyStoreInvalidStateException`, `KMSInvalidStateException`

Certains proxys de magasin de clés externe mettent en œuvre des exigences d'autorisation pour l'utilisation de leurs clés externes. Un proxy de magasin de clés externe est autorisé, mais pas tenu, de concevoir et d'implémenter un schéma d'autorisation qui permet à des utilisateurs particuliers de demander des opérations particulières sous certaines conditions. Par exemple, un proxy peut autoriser un utilisateur à chiffrer avec une clé externe particulière, mais pas à déchiffrer avec elle. Pour plus d’informations, consultez Autorisation par proxy de magasin de clés externe (facultatif).

L'autorisation du proxy est basée sur les métadonnées qu'AWS KMS inclut dans ses requêtes au proxy. Les champs `awsSourceVpc` et `awsSourceVpce` ne sont inclus dans les métadonnées que lorsque la requête provient d'un point de terminaison d'un VPC et uniquement lorsque l'appelant possède le même compte que la clé KMS.

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

Lorsque le proxy rejette une requête en raison d'un échec d'autorisation, l'opération AWS KMS correspondante échoue. L'opération `CreateKey` renvoie `CustomKeyStoreInvalidStateException` et les opérations de chiffrement AWS KMS renvoient `KMSInvalidStateException`. Toutes deux utilisent le message d'erreur suivant :

> Le proxy de magasin de clés externe a refusé l'accès à l'opération. Vérifiez que l'utilisateur et la clé externe sont tous deux autorisés pour cette opération, puis réitérez la requête.

- Pour résoudre l'erreur, utilisez votre gestionnaire de clés externe ou les outils de proxy de votre magasin de clés externe pour déterminer la raison de l'échec de l'autorisation. Ensuite, mettez à jour la procédure à l'origine de la requête non autorisée ou utilisez les outils de proxy de votre magasin de clés externe pour mettre à jour la politique d'autorisation. Vous ne pouvez pas résoudre cette erreur dans AWS KMS.

Référence des types de clés

AWS KMS prend en charge différentes fonctions pour différents types de clés KMS. Par exemple, vous ne pouvez utiliser que des clés KMS de chiffrement symétriques pour générer des clés de données symétriques et des paires de clés de données asymétriques. En outre, l'importation d'un élément de clé et la rotation automatique des clés sont prises en charge uniquement pour les clés KMS de chiffrement symétriques. Vous pouvez créer uniquement des clés KMS de chiffrement symétriques dans un magasin de clés personnalisé.

Cette référence comprend deux tableaux.

- Le tableau des types de clé répertorie les opérations AWS KMS qui sont valides pour les clés KMS de chiffrement symétriques, les clés KMS asymétriques et les clés KMS HMAC.
- Le tableau de fonctionnalités spéciales répertorie les opérations AWS KMS qui sont valides pour les clés KMS multi-régions, les clés KMS avec des éléments de clé importés et les clés KMS des magasins de clés personnalisés.

Tableau des types de clé

Il peut être nécessaire de faire défiler horizontalement ou verticalement pour afficher toutes les données de ce tableau.

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
CancelKeyDeletion	✓	✓	✓	✓
CreateAlias	✓	✓	✓	✓
CreateGrant	✓	✓	✓	✓
CreateKey	✓	✓	✓	✓
Decrypt	✓	⊗	✓	⊗
DeleteAlias	✓	✓	✓	✓
DeleteImportedKeyMaterial	✓	✓	✓	✓

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
Valable uniquement sur les clés KMS avec des éléments de clé importés (`Origin` est `EXTERNAL`).				
DescribeKey	✓	✓	✓	✓
DisableKey	✓	✓	✓	✓
DisableKeyRotation	✓ Valable uniquement sur les clés KMS avec l'élément de clé AWS KMS (`Origin` est `AWS_KMS`).	⊗	⊗	⊗
EnableKey	✓	✓	✓	✓

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
EnableKeyRotation	✓ Valable uniquement sur les clés KMS avec l'élément de clé AWS KMS (`Origin` est `AWS_KMS`).	⊗	⊗	⊗
Encrypt	✓	⊗	✓	⊗
GenerateDataKey	✓	⊗	⊗	⊗
GenerateDataKeyPair Génère une paire de clés de données asymétriques qui est protégée par une clé KMS de chiffrement symétrique.	✓ Non valable sur les clés KMS des magasins de clés personnalisés.	⊗	⊗	⊗

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
GenerateDataKeyPairWithoutPlaintext Génère une paire de clés de données asymétriques qui est protégée par une clé KMS de chiffrement symétrique.	✓ Non valable sur les clés KMS des magasins de clés personnalisés.	⊗	⊗	⊗
GenerateDataKeyWithoutPlaintext	✓	⊗	⊗	⊗
GenerateMac	⊗	✓	⊗	⊗
GetKeyPolicy	✓	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` sera toujours `false`.)	✓ (`KeyRotationEnabled` sera toujours `false`.)	✓ (`KeyRotationEnabled` sera toujours `false`.)

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
GetParametersForImport Valable uniquement sur les clés KMS avec des éléments de clé importés (`Origin` est `EXTERNAL`).	✓	✓	✓	✓
GetPublicKey	⊗	⊗	✓	✓
ImportKeyMaterial Valable uniquement sur les clés KMS avec des éléments de clé importés (`Origin` est `EXTERNAL`).	✓	✓	✓	✓
ListAliases	✓	✓	✓	✓
ListGrants	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓
ListResourceTags	✓	✓	✓	✓
ListRetirableGrants	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓
ReEncrypt	✓	⊗	✓	⊗

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
ReplicateKey - Valide uniquement sur les clés multi-région	✓	✓	✓	✓
RetireGrant	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓	✓
Sign (Signer)	⊗	⊗	⊗	✓
TagResource	✓	✓	✓	✓
UntagResource	✓	✓	✓	✓
UpdateAlias La clé KMS actuelle et la nouvelle clé KMS doivent être du même type (toutes deux soit symétriques, soit asymétriques, soit HMAC) et avoir la même utilisation de clé.	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	✓	✓

Opération d'API AWS KMS	Clés KMS de chiffrement symétriques	Clés KMS HMAC	Clés KMS asymétriques (ENCRYPT_DECRYPT)	Clés KMS asymétriques (SIGN_VERIFY)
UpdateReplicaRegion - Valide uniquement sur les clés multi-région	✓	✓	✓	✓
Verify	⊗	⊗	⊗	✓
VerifyMac	⊗	✓	⊗	⊗

Tableau des fonctionnalités spéciales

Ce tableau présente les opérations d'API AWS KMS prises en charge sur chaque type de clé à usage spécial.

En lisant ce tableau, soyez attentif aux interactions suivantes :

- Clés multi-région:
  - Les clés multi-régions peuvent être des clés KMS de chiffrement symétriques, des clés KMS asymétriques, des clés KMS HMAC et des clés KMS avec éléments de clé importés.
  - Vous ne pouvez pas créer de clés multi-région dans un magasin de clés personnalisé.
- Éléments de clé importés
  - Vous pouvez importer des éléments de clé pour des clés KMS de chiffrement symétriques, des clés KMS asymétriques et des clés KMS HMAC.
  - Vous pouvez créer des clés multi-région avec des éléments de clé importés.
  - Vous ne pouvez pas créer de clés avec des éléments de clé importés dans un magasin de clés personnalisé.
  - La rotation automatique des clés (`EnableKeyRotation`, `DisableKeyRotation`) n'est pas prise en charge pour les clés KMS avec des éléments de clé importés.
- Magasins de clés personnalisés

- Les magasins de clés personnalisés ne prennent en charge que les clés KMS de chiffrement symétriques.
- Les opérations symétriques sur des paires de clés asymétriques (`GenerateDataKeyPair`, `GenerateDataKeyPairWithoutPlaintext`) ne sont pas prises en charge sur les clés KMS dans les magasins de clés personnalisés.
- La rotation automatique des clés (`EnableKeyRotation`, `DisableKeyRotation`) n'est pas prise en charge sur les clés KMS dans les magasins de clés personnalisés.
- Vous ne pouvez pas créer de clés multi-région dans les magasins de clés personnalisés.

Il peut être nécessaire de faire défiler horizontalement ou verticalement pour afficher toutes les données de ce tableau.

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey Vous pouvez utiliser `CreateKey` pour créer une clé primaire multi-régions, une clé KMS avec des éléments de clé importés ou une clé KMS dans un magasin de clés personnalisé. Pour créer une clé de réplica multi-régions, utilisez `ReplicateKey` .	✓	✓	✓
Decrypt	✓	✓	✓

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
	Valable uniquement lorsque `KeyUsage` est `ENCRYPT_DECRYPT`		
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓ Valable uniquement pour les clés avec éléments de clé importés (`Origin` est `EXTERNAL`)	✓	⊗
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
DisableKeyRotation	✓ Valable uniquement sur les clés de chiffrement symétrique avec éléments de clé AWS KMS (`Origin` est `AWS_KMS`)	⊗	⊗
EnableKey	✓ Valable uniquement sur les clés KMS de chiffrement symétrique	✓	✓
EnableKeyRotation	✓ Valable uniquement sur les clés de chiffrement symétrique avec éléments de clé AWS KMS (`Origin` est `AWS_KMS`)	⊗	⊗

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
Encrypt	✓ Valable uniquement lorsque `KeyUsage` est `ENCRYPT_DECRYPT`	✓	✓
GenerateDataKey	✓ Valable uniquement sur les clés KMS de chiffrement symétrique	✓	✓
GenerateDataKeyPair	✓ Valable uniquement sur les clés KMS de chiffrement symétrique	✓	⊗

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
GenerateDataKeyPairWithoutPlaintext	✓ Valable uniquement sur les clés KMS de chiffrement symétrique	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ Valable uniquement sur les clés KMS de chiffrement symétrique	✓	✓
GenerateMac Valable uniquement sur les clés KMS HMAC	✓	✓	⊗
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` sera toujours `false`.)	⊗

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
GetParametersForImport	✓ Valable uniquement pour les clés avec éléments de clé importés (`Origin` est `EXTERNAL`)	✓	⊗
GetPublicKey Valable uniquement pour les clés KMS asymétriques	✓	✓	⊗
ImportKeyMaterial	✓ Valable uniquement pour les clés avec éléments de clé importés (`Origin` est `EXTERNAL`)	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ Valable uniquement lorsque `KeyUsage` est `ENCRYPT_DECRYPT`	✓	✓
ReplicateKey	✓ Valable uniquement sur les clés primaires multi-régions	✓ Valable uniquement sur les clés primaires multi-régions	⊗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
Sign (Signer) Valable uniquement lorsque `KeyUsage` est `SIGN_VERIFY`	✓	✓	⊗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓
UpdateAlias – La clé KMS actuelle et la nouvelle clé KMS doivent être du même type (toutes deux soit symétriques, soit asymétriques, soit HMAC) et avoir la même utilisation de clé.	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓ Valable uniquement sur les clés multi-régions	⊗
Vérification Valide uniquement lorsque `KeyUsage` est `SIGN_VERIFY` .	✓	✓	⊗

Opération d'API AWS KMS	Clés multi-région	Éléments de clé importés	Clés KMS dans un magasin de clés personnalisé
VerifyMac Valable uniquement sur les clés KMS HMAC	✓	✓	⊗

Sécurité de AWS Key Management Service

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous-même. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le cloud AWS. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformité AWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Key Management Service (AWS KMS), veuillez consulter AWS Services concernés par le programme de conformité.
- Sécurité dans le cloud : votre responsabilité est déterminée par le service AWS que vous utilisez. Dans AWS KMS, en plus de votre configuration et de votre utilisation des AWS KMS keys, vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS Key Management Service. Elle vous montre comment configurer AWS KMS pour atteindre vos objectifs en matière de sécurité et de conformité.

Rubriques

- Protection des données dans AWS Key Management Service
- Gestion des identités et des accès pour AWS Key Management Service
- Journalisation et surveillance dans AWS Key Management Service
- Validation de la conformité pour AWS Key Management Service
- Résilience dans AWS Key Management Service
- Sécurité de l'infrastructure dans AWS Key Management Service
- Bonnes pratiques de sécurité pour AWS Key Management Service

Protection des données dans AWS Key Management Service

AWS Key Management Service stocke et protège vos clés de chiffrement pour les rendre hautement disponibles tout en vous offrant un contrôle d'accès solide et flexible.

Rubriques

- Protection des éléments de clé
- Chiffrement des données
- Confidentialité du trafic inter-réseaux

Protection des éléments de clé

Par défaut, AWS KMS génère et protège l'élément de clé cryptographique pour les clés KMS. En outre, AWS KMS offre des options pour l'élément de clé créé et protégé à l'extérieur de AWS KMS. Pour plus de détails techniques sur les clés KMS et les éléments de clé, veuillez consulter Détails de chiffrement AWS Key Management Service.

Protection de l'élément de clé généré dans AWS KMS

Lorsque vous créez une clé KMS, par défaut, AWS KMS génère et protège les éléments de chiffrement pour cette clé KMS.

Pour protéger les éléments de clé pour les clés KMS, AWS KMS s'appuie sur une flotte distribuée de modules de sécurité matériels (HSM) validés par la norme FIPS 140-2, niveau de sécurité 3. Chaque HSM AWS KMS est un dispositif matériel autonome conçu pour fournir des fonctions de chiffrement dédiées, afin de répondre aux exigences de sécurité et de capacité de mise à l'échelle de AWS KMS. (Les HSM que AWS KMS utilise dans les régions de la Chine sont conformes à la réglementation OSCCA et à toutes les réglementations chinoises pertinentes, mais ne sont pas validés conformément au Programme de validation des modules de chiffrement FIPS 140-2.)

L'élément de clé d'une clé KMS est chiffré par défaut lorsqu'il est généré dans le HSM. L'élément de clé est déchiffré uniquement dans la mémoire volatile du HSM et uniquement pendant les quelques millisecondes nécessaires pour l'utiliser dans une opération cryptographique. Chaque fois que l'élément de clé n'est pas utilisé activement, il est chiffré dans le HSM et transféré vers un stockage persistant hautement durable (99,999999999 %) et à faible latence, où il reste séparé et isolé des HSM. Les éléments de clé en texte clair ne quittent jamais les limites de sécurité du HSM ; ils ne sont jamais écrits sur disque ou conservés sur un support de stockage. (La seule exception est la clé publique d'une paire de clés asymétriques, qui n'est pas secrète.)

AWS affirme comme principe de sécurité fondamental qu'il n'y a aucune interaction humaine avec les clés cryptographiques en texte clair de quelque type que ce soit dans un Service AWS. Il n'existe aucun mécanisme permettant à quiconque, y compris aux opérateurs Service AWS, de visualiser, d'accéder ou d'exporter un élément de clé en texte brut. Ce principe s'applique même lors de défaillances catastrophiques et d'événements de reprise après sinistre. L'élément de clé client en texte clair dans AWS KMS est utilisé pour les opérations cryptographiques dans les HSM validés par la norme FIPS AWS KMS uniquement en réponse aux demandes autorisées adressées au service par le client ou son délégué.

Pour les clés gérées par le client, le Compte AWS qui crée la clé est le propriétaire unique et non transférable de la clé. Le compte propriétaire a un contrôle complet et exclusif sur les politiques d'autorisation qui contrôlent l'accès à la clé. Pour Clés gérées par AWS, le Compte AWS a un contrôle total sur les politiques IAM qui autorisent les demandes adressées au Service AWS.

Protection de l'élément de clé généré à l'extérieur de AWS KMS

AWS KMS fournit des alternatives à l'élément de clé généré dans AWS KMS.

Les magasins de clés personnalisés, une fonction AWS KMS facultative, vous permettent de créer des clés KMS soutenues par des éléments de clé générés et utilisés en dehors deAWS KMS. Les clés KMS des magasins de clés AWS CloudHSM sont soutenues par des clés des modules de sécurité matériels AWS CloudHSM que vous contrôlez. Ces HSM sont certifiés selon la norme FIPS 140-2 de niveau de sécurité 3. Les clés KMS des magasins de clés externes sont soutenues par des clés d'un gestionnaire de clés externe que vous contrôlez et gérez en dehors de AWS, tel qu'un HSM physique dans votre centre de données privé.

Une autre fonction facultative vous permet d'importer des éléments de clé pour une clé KMS. Pour protéger l'élément de clé importé pendant son transport vers AWS KMS, vous devez le chiffrer à l'aide d'une clé publique issue d'une paire de clés RSA générée dans un HSM AWS KMS. Les éléments de clé importés sont déchiffrés sur un HSM AWS KMS puis rechiffrés sous des clés symétriques dans le HSM. Comme tous les éléments de clé AWS KMS, les éléments de clé importés en texte clair ne quittent jamais les HSM non chiffrés. Cependant, le client qui a fourni les éléments de clé est responsable de l'utilisation en toute sécurité, de la durabilité et de la maintenance des éléments de clé en dehors de AWS KMS.

Chiffrement des données

Les données dans AWS KMS comprennent : les AWS KMS keys et les éléments de clé de chiffrement qu'ils représentent. Ces éléments de clé existent en texte clair uniquement dans des

modules de sécurité matérielle (HSM) AWS KMS et uniquement lors de l'utilisation. Sinon, les éléments de clé sont chiffrés et stockés dans un stockage permanent durable.

Les éléments de clé générés par AWS KMS pour les clés KMS ne quittent jamais les limites des HSM AWS KMS non chiffrés. Ils ne sont ni exportés, ni transmis dans des opérations d'API AWS KMS. L'exception concerne les clés multi-régions, où AWS KMS utilise un mécanisme de réplication entre régions pour copier l'élément de clé d'une clé multi-régions d'un HSM d'une Région AWS à un HSM d'une autre Région AWS. Pour plus d'informations, veuillez consulter la rubrique Replication process for multi-Region keys (Processus de réplication pour clés multi-régions) dans AWS Key Management Service Cryptographic Details (Détails de chiffrement).

Rubriques

- Chiffrement au repos
- Chiffrement en transit

Chiffrement au repos

AWS KMS génère des éléments de clé pour AWS KMS keys dans des modules de sécurité matérielle (HSM) conformes aux normes FIPS 140-2 niveau de sécurité 3. La seule exception concerne les régions de Chine, où les HSM AWS KMS utilisées pour générer des clés KMS sont conformes à toutes les réglementations chinoises pertinentes, mais ne sont pas validées dans le cadre du programme de validation FIPS 140-2 du module cryptographique. Lorsqu'ils ne sont pas utilisés, les éléments de clé sont chiffrés par une clé HSM et écrits dans un stockage permanent et persistant. Les éléments de clé pour les clés KMS et les clés de chiffrement qui protègent les éléments de clé ne quittent jamais les HSM sous forme de texte brut.

Le chiffrement et la gestion des éléments de clé pour les clés KMS sont entièrement gérés par AWS KMS.

Pour plus d'informations, veuillez consulter Utilisation des AWS KMS keys dans les Détails de chiffrement AWS Key Management Service.

Chiffrement en transit

Les éléments de clé générés par AWS KMS pour les clés KMS ne sont jamais exportés ou transmis vers des opérations d'API AWS KMS. AWS KMS utilise des identificateurs de clé pour représenter les clés KMS dans les opérations d'API. De même, les éléments de clé pour les clés KMS dans les magasins de clés personnalisés AWS KMS ne sont pas exportables et ne sont jamais transmis dans AWS KMS ou dans des opérations d'API AWS CloudHSM.

Cependant, certaines opérations d'API AWS KMS renvoient des clés de données. En outre, les clients peuvent utiliser les opérations d'API pourimporter des éléments de clé pour les clés KMS sélectionnées.

Tous les appels d'API AWS KMS doivent être signés et transmis à l'aide du protocole TLS (Transport Layer Security). AWS KMS nécessite TLS 1.2 et recommande TLS 1.3 dans toutes les régions. AWS KMS prend également en charge le protocole TLS post-quantique hybride pour les points de terminaison de service AWS KMS dans toutes les régions, à l'exception des régions chinoises. AWS KMS ne prend pas en charge le protocole TLS post-quantique hybride pour les points de terminaison FIPS dans AWS GovCloud (US). Les appels vers AWS KMS nécessitent également une suite de chiffrement moderne qui prend en charge une confidentialité persistante parfaite, ce qui signifie que toute violation de secret, telle qu'une clé privée, ne compromet pas également la clé de session.

Si vous avez besoin de modules cryptographiques validés FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information Processing Standard). Pour utiliser les points de terminaison AWS KMS standard ou les points de terminaison FIPS AWS KMS, les clients doivent prendre en charge le protocole TLS 1.2 ou une version ultérieure. Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale). Pour obtenir la liste complète des points de terminaison FIPS AWS KMS, consultez Points de terminaison et quotas de AWS Key Management Service dans le Références générales AWS.

Les communications entre les hôtes de service AWS KMS et les HSM sont protégées grâce au chiffrement de courbe elliptique (ECC) et à la norme de chiffrement avancée (AES) dans un dispositif de chiffrement authentifié. Pour plus d'informations, veuillez consulter Sécurité des communications internes dans Détails de chiffrement AWS Key Management Service.

Confidentialité du trafic inter-réseaux

AWS KMS prend en charge une AWS Management Console et un ensemble d'opérations d'API vous permettant de créer et de gérer des AWS KMS keys et de les utiliser dans des opérations de chiffrement.

AWS KMS prend en charge deux options de connectivité réseau depuis votre réseau privé vers AWS.

- Une onnexion VPN IPsec via Internet

- AWS Direct Connect relie votre réseau interne à un emplacement AWS Direct Connect via un câble en fibres optiques Ethernet standard.

Tous les appels d'API AWS KMS doivent être signés et transmis à l'aide du protocole TLS (Transport Layer Security). Les appels nécessitent également une suite de chiffrement moderne qui prend en charge une confidentialité persistante et parfaite. Le trafic vers les HSM qui stockent les éléments de clé pour les clés KMS est autorisé uniquement à partir des hôtes d'API AWS KMS connus via le réseau interne AWS.

Pour se connecter directement à AWS KMS depuis votre Virtual Private Cloud (VPC) sans envoyer de trafic sur le réseau Internet public, utilisez des points de terminaison d'un VPC, optimisés par AWS PrivateLink. Pour plus d'informations, consultez Connexion à AWS KMS via un point de terminaison d'un VPC.

AWS KMS prend également en charge une option d'échange de clés post-quantiques hybrides pour le protocole de chiffrement réseau TLS (Transport Layer Security). Vous pouvez utiliser cette option avec TLS lorsque vous vous connectez aux points de terminaison de l'API AWS KMS.

Gestion des identités et des accès pour AWS Key Management Service

AWS Identity and Access Management (IAM) vous permet de contrôler en toute sécurité l'accès aux ressources AWS. Les administrateurs déterminent qui peut s'authentifier (se connecter) et être autorisé (disposer des autorisations) à utiliser des ressources AWS KMS. Pour plus d'informations, consultez Utilisation des politiques IAM avec AWS KMS.

Les stratégies de clé constituent le principal moyen de contrôler l'accès aux clés KMS dans AWS KMS. Chaque clé KMS doit avoir une politique de clé. Vous pouvez également utiliser des stratégies IAM et des octrois, ainsi que des stratégies de clé pour contrôler l'accès à vos clés KMS. Pour plus d'informations, consultez Authentification et contrôle d'accès pour AWS KMS.

Si vous utilisez un Amazon Virtual Private Cloud (Amazon VPC), vous pouvez créer un point de terminaison d'un VPC d'interface sur AWS KMS optimisé par AWS PrivateLink. Vous pouvez également utiliser des stratégies de point de terminaison d'un VPC pour déterminer quels mandataires peuvent accéder à votre point de terminaison AWS KMS, les appels d'API qu'ils peuvent effectuer et la clé KMS à laquelle ils peuvent accéder. Pour plus d'informations, consultez Contrôle de l'accès à votre point de terminaison d'un VPC.

Journalisation et surveillance dans AWS Key Management Service

La surveillance est un élément important permettant de comprendre la disponibilité, l'état et l'utilisation de vos AWS KMS keys dans AWS KMS. La surveillance permet d'assurer la sécurité, la fiabilité, la disponibilité et les performances de vos solutions AWS. AWS fournit plusieurs outils pour surveiller vos clés KMS.

Journaux AWS CloudTrail

Chaque appel à une opération d'API AWS KMS est capturé comme événement dans un journal AWS CloudTrail. Ces journaux enregistrent tous les appels d'API à partir de la console AWS KMS, ainsi que les appels effectués par AWS KMS et d'autres services AWS. Les appels d'API entre comptes, tels qu'un appel à utiliser une clé KMS dans un autre compteCompte AWS, sont enregistrés dans les CloudTrail journaux des deux comptes.

Lors du dépannage ou de l'audit, vous pouvez utiliser le journal pour reconstruire le cycle de vie d'une clé KMS. Vous pouvez également afficher sa gestion et son utilisation de la clé KMS dans les opérations de chiffrement. Pour plus d’informations, consultez the section called “Journalisation avec AWS CloudTrail”.

Amazon CloudWatch Logs

Contrôlez vos fichiers journaux, stockez-les et accédez-y à partir d'AWS CloudTrail ou d'autres sources. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

CarAWS KMS, CloudWatch stocke des informations utiles qui vous aident à éviter les problèmes liés à vos clés KMS et aux ressources qu'elles protègent. Pour plus d’informations, consultez the section called “Surveillance avec CloudWatch”.

Amazon EventBridge

AWS KMSgénère EventBridge des événements lorsque votre clé KMS est pivotée ou supprimée ou lorsque le contenu clé importé de votre clé KMS expire. Recherchez les événements AWS KMS (opérations d'API) et acheminez-les vers un ou plusieurs fonctions ou flux cibles pour capturer les informations de l'état. Pour plus d'informations, consultez the section called “Surveillance avec Amazon EventBridge” le guide de EventBridge l'utilisateur Amazon.

CloudWatch Métriques Amazon

Vous pouvez surveiller vos clés KMS à l'aide de CloudWatch métriques, qui collectent et traitent les données brutes pour AWS KMS en faire des indicateurs de performance. Les données sont

enregistrées à intervalles de deux semaines afin que vous puissiez visualiser les tendances des informations actuelles et historiques. Cela vous aide à comprendre comment vos clés KMS sont utilisées et comment leur utilisation évolue au fil du temps. Pour plus d'informations sur l'utilisation de CloudWatch métriques pour surveiller les clés KMS, consultezMétriques et dimensions AWS KMS.

CloudWatch Alarmes Amazon

Surveillez les évolutions d'une seule métrique pendant la période que vous spécifiez. Ensuite, prenez des mesures en fonction de la valeur de la métrique par rapport à un seuil sur un certain nombre de périodes. Par exemple, vous pouvez créer une CloudWatch alarme qui se déclenche lorsque quelqu'un essaie d'utiliser une clé KMS dont la suppression est programmée dans le cadre d'une opération cryptographique. Cela indique que la clé KMS est toujours utilisée et ne devrait probablement pas être supprimée. Pour plus d’informations, consultez the section called “Création d'une alarme”.

AWS Security Hub

Vous pouvez surveiller votre utilisation AWS KMS par rapport aux normes et aux meilleures pratiques de l'industrie de la sécurité à l'aide de AWS Security Hub. Security Hub utilise des contrôles de sécurité pour évaluer les configurations des ressources et les normes de sécurité afin de vous aider à respecter divers cadres de conformité. Pour plus d'informations, consultez Concepts AWS Key Management Service dans le Guide de l'utilisateur AWS Security Hub.

Validation de la conformité pour AWS Key Management Service

Les auditeurs tiers évaluent la sécurité et la conformité de AWS Key Management Service dans le cadre de plusieurs programmes de conformité AWS. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et autres.

Rubriques

- Documents de conformité et de sécurité
- En savoir plus

Documents de conformité et de sécurité

Les documents suivants liés à la sécurité et à la conformité couvrent AWS KMS. Pour ce faire, utilisez AWS Artifact.

- Cloud Computing Compliance Controls Catalogue (C5)
- ISO 27001 : Déclaration d'applicabilité 2013 (DdA)
- ISO 27001 : Certification 2013
- ISO 27017 : Déclaration d'applicabilité 2015 (DdA)
- ISO 27017 : Certification 2015
- ISO 27018 : Déclaration d'applicabilité 2015 (DdA)
- ISO 27018 : Certification 2014
- ISO 9001: Certification 2015
- Attestation de conformité (AOC) et récapitulatif des responsabilités PCI DSS
- Rapport SOC 1 (Service Organization Controls)
- Rapport SOC 2 (Service Organization Controls)
- Rapport SOC 2 (Service Organization Controls) relatif à la confidentialité
- FedRAMP-High

Pour obtenir une aide relative à l'utilisation de AWS Artifact, veuillez consulter Téléchargement des rapports dans AWS Artifact.

En savoir plus

Votre responsabilité de conformité lors de l'utilisation de AWS KMS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables. Si votre utilisation de AWS KMS est soumise à la conformité aux normes publiées, AWS fournit des ressources pour vous aider :

- Services AWS concernés par le programme de conformité – Cette page répertorie les services AWS concernés par les programmes de conformité spécifiques. Pour obtenir des informations générales, consultez Programmes de conformité AWS.
- Guides Quick Start de la sécurité et de la conformité : ces guides de déploiement traitent de considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence centrés sur la sécurité et la conformité dans AWS.
- Ressources de conformité AWS : cet ensemble de manuels et de guides peut s’appliquer à votre secteur et à votre emplacement.
- AWS Config : ce service AWS permet d’évaluer la conformité des configurations de vos ressources par rapport à des pratiques internes, règlementations et autres directives sectorielles.

- AWS Security Hub – Ce service AWS fournit une vue complète de votre état de sécurité à l'intérieur de AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.

Résilience dans AWS Key Management Service

L'infrastructure mondiale d'AWS est construite autour de zones de disponibilité et de Régions AWS. Les Régions AWSfournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Outre l'infrastructure globale AWS, AWS KMS propose plusieurs fonctionnalités qui contribuent à la prise en charge des vos besoins en matière de résilience et de sauvegarde de données. Pour plus d'informations sur les Régions AWS et les zones de disponibilité, consultez Infrastructure mondiale d'AWS.

Isolement régional

AWS Key Management Service (AWS KMS) est un service régional autonome disponible dans tous les Régions AWS. La conception d'Isolement régional de AWS KMS garantit qu'un problème de disponibilité dans une Région AWS ne peut affecter l'opération AWS KMS dans n'importe quelle autre région. AWS KMS est conçu pour garantir un temps d'interruption planifié nul, avec toutes les mises à jour logicielles et les opérations de mise à l'échelle effectuées de manière transparente et imperceptible.

LeAWS KMS Contrat de niveau de service (SLA) inclut un engagement de service de 99,999 % pour toutes les API KMS. Pour remplir cet engagement, AWS KMS garantit que toutes les données et informations d'autorisation nécessaires à l'exécution d'une requête d'API sont disponibles sur tous les hôtes régionaux qui reçoivent la requête.

L'infrastructure AWS KMS est répliquée dans au moins trois zones de disponibilité (AZ) dans chaque région. Pour s'assurer que les défaillances de plusieurs hôtes n'affectent pas les performances de

AWS KMS, AWS KMS est conçu pour traiter le trafic client provenant de n'importe quelle zone de disponibilité d'une région.

Les modifications que vous apportez aux propriétés ou aux autorisations d'une clé KMS sont répliquées pour tous les hôtes de la région afin de garantir que la requête ultérieure peut être traitée correctement par n'importe quel hôte de la région. Les requêtes d'opérations de chiffrement à l'aide de votre clé KMS sont réacheminées vers une flotte de modules de sécurité matérielle (HSM) AWS KMS, dont n'importe lequel peut effectuer l'opération avec la clé KMS.

Conception à locataires multiples

La conception à locataires multiples de AWS KMS lui permet de respecter le contrat de niveau de service (SLA) de disponibilité de 99,999 % et de maintenir des taux de demande élevés, tout en protégeant la confidentialité de vos clés et données.

Plusieurs mécanismes renforçant l'intégrité sont déployés pour garantir que la clé KMS que vous avez spécifiée pour l'opération cryptographique est toujours celle utilisée.

Les éléments de clé en texte clair de vos clés KMS sont protégés de manière stricte. Les éléments de clé sont chiffrés dans le HSM dès sa création, et les éléments de clé chiffrés sont immédiatement déplacés vers un stockage sécurisé à faible latence. La clé chiffrée est récupérée et déchiffrée dans le HSM juste à temps pour être utilisée. La clé en texte clair reste dans la mémoire HSM uniquement pendant le temps nécessaire à la réalisation de l'opération cryptographique. Il est ensuite rechiffré dans le HSM et la clé chiffrée est renvoyée au stockage. Les éléments de clé en texte clair ne quittent jamais les HSM ; ils ne sont jamais écrits dans un stockage persistant.

Pour plus d'informations sur les mécanismes utilisés par AWS KMS pour sécuriser vos clés, voir Détails cryptographiques d'AWS Key Management Service.

Bonnes pratiques de résilience dans AWS KMS

Pour optimiser la résilience pour vos ressources AWS KMS, envisagez les stratégies suivantes.

- Pour prendre en charge votre stratégie de sauvegarde et de reprise après sinistre, prenez en compte les Clés multi-régions, qui sont des clés KMS créées dans une Région AWS et répliquées uniquement pour les régions que vous spécifiez. Avec les clés multi-régions, vous pouvez déplacer des ressources chiffrées entre Régions AWS (dans la même partition) sans jamais exposer le texte clair, et déchiffrer la ressource, si nécessaire, dans l'une de ses régions de destination. Les clés multi-régions associées sont interopérables car elles partagent les mêmes éléments de clé et le

même ID de clé, mais elles disposent de stratégies clé indépendantes pour le contrôle d'accès haute résolution. Pour plus de détails, veuillez consulter Clés multi-régions dans AWS KMS.

- Pour protéger vos clés dans un service à locataires multiples comme AWS KMS, assurez-vous d'utiliser les contrôles d'accès, y compris Stratégies de clé et Stratégies IAM. En outre, vous pouvez envoyer vos requêtes à AWS KMS à l'aide d'un point de terminaison d'interface VPC optimisé par AWS PrivateLink. Dans ce cas, toutes les communications entre votre VPC Amazon et AWS KMS sont gérées entièrement dans le réseau AWS, en utilisant un point de terminaison dédié AWS KMS limité à votre VPC. Vous pouvez sécuriser davantage ces requêtes en créant une couche d'autorisation supplémentaire à l'aide des Stratégies de point de terminaison d'un VPC. Pour en savoir plus, consultez la section Connexion à AWS KMS via un point de terminaison d'un VPC.

Sécurité de l'infrastructure dans AWS Key Management Service

En tant que service géré, AWS Key Management Service (AWS KMS) est protégé par les procédures de sécurité du réseau mondial AWS qui sont décrites dans le livre blanc Amazon Web Services : Présentation des procédures de sécurité.

Pour accéder à AWS KMS via le réseau, vous pouvez appeler les opérations d'API AWS KMS qui sont décrites dans la Référence d'API AWS Key Management Service. AWS KMS nécessite TLS 1.2 et recommande TLS 1.3 dans toutes les régions. AWS KMS prend également en charge le protocole TLS post-quantique hybride pour les points de terminaison de service AWS KMS dans toutes les régions, à l'exception des régions chinoises. AWS KMS ne prend pas en charge le protocole TLS post-quantique hybride pour les points de terminaison FIPS dans AWS GovCloud (US). Pour utiliser les points de terminaison AWS KMS standard ou les points de terminaison FIPS AWS KMS, les clients doivent prendre en charge le protocole TLS 1.2 ou une version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes telles que Java 7 et versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Vous pouvez appeler ces opérations d'API à partir de n'importe quel endroit du réseau, mais AWS KMS prend en charge les conditions de stratégie globales qui vous permettent de contrôler l'accès à

une clé KMS en fonction de l'adresse IP source, du VPC et du point de terminaison d'un VPC. Vous pouvez utiliser ces clés de condition dans les stratégies de clé et les stratégies IAM. Cependant, ces conditions peuvent empêcher AWS d'utiliser la clé KMS en votre nom. Pour plus de détails, consultez AWS clés de condition globales.

Par exemple, l'instruction de politique de clé suivante autorise les utilisateurs qui peuvent endosser le rôle `KMSTestRole` à utiliser cette AWS KMS key pour les opérations de chiffrement spécifiées, à moins que l'adresse IP source ne soit l'une des adresses IP spécifiées dans la politique.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS":
    "arn:aws:iam::111122223333:role/KMSTestRole"},
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24",
          "203.0.113.0/24"
        ]
      }
    }
  }
}
```

Isolation des hôtes physiques

La sécurité de l'infrastructure physique utilisée par AWS KMS est soumise aux contrôles décrits dans la section Sécurité physique et environnementale du livre blanc Amazon Web Services : Présentation des processus de sécurité. Vous trouverez plus de détails dans les rapports de conformité et les résultats d'audit tiers répertoriés dans la section précédente.

AWS KMS est pris en charge par des modules de sécurité matériels (HSM) dédiés conçus avec des contrôles spécifiques pour résister aux attaques physiques. Les HSM sont des périphériques physiques qui ne possèdent pas de couche de virtualisation, telle qu'un hyperviseur, qui partage le périphérique physique entre plusieurs locataires logiques. Les éléments de clé des AWS KMS keys sont stockés uniquement dans la mémoire volatile des HSM, et uniquement durant l'utilisation de la clé KMS. Cette mémoire est effacée lorsque le HSM sort de l'état opérationnel, y compris lors des arrêts et des réinitialisations prévus et involontaires. Pour obtenir des informations détaillées sur le fonctionnement des HSM AWS KMS, veuillez consulter AWS Key Management Service Détails cryptographiques.

Bonnes pratiques de sécurité pour AWS Key Management Service

AWS Key Management Service (AWS KMS) prend en charge de nombreuses fonctions de sécurité que vous pouvez implémenter pour améliorer la protection de vos clés de chiffrement, notamment les politiques de clé et les politiques IAM, une option de contexte de chiffrement pour les opérations cryptographiques sur les clés de chiffrement symétriques, un ensemble complet de clés de condition pour affiner vos politiques de clé et IAM, et des contraintes d'octroi pour limiter les octrois.

Ces fonctions de sécurité sont décrites en détail dans le document Bonnes pratiques AWS Key Management Service (PDF). Les directives générales de ce livre blanc ne font pas office de solution de sécurité complète. Étant donné que toutes les bonnes pratiques ne conviennent pas à toutes les situations, elles ne sont pas censées être prescriptives.

Voir aussi

- Bonnes pratiques pour les politiques IAM
- Bonnes pratiques relatives aux octrois AWS KMS
- Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM

Quotas

Pour rendre AWS KMS réactif et performant pour tous les utilisateurs, AWS KMS applique deux types de quotas : des quotas de ressources et des quotas de demandes. Chaque quota est calculé indépendamment pour chaque région de chaque Compte AWS.

Tous les quotas AWS KMS sont ajustables, à l'exception du quota de ressources de la taille du document de politique de clé et du AWS CloudHSMquota de ressources des magasins de clés personnalisés. Pour demander une augmentation de quota, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas. Pour demander une réduction de quota, pour modifier un quota qui ne figure pas dans la liste des Service Quotas, ou pour modifier un quota dans une Région AWS où les Service Quotas pour AWS KMS ne sont pas disponibles, rendez-vous dans le Centre AWS Support et créez un dossier.

Rubriques

- Quotas de ressources
- Quotas de demande
- Limitation des demandes AWS KMS

Quotas de ressources

AWS KMS établit des quotas de ressources afin d'être sûr de pouvoir fournir un service rapide et résilient à tous nos clients. Certains quotas de ressources s'appliquent uniquement aux ressources que vous créez, mais pas aux ressources que les services AWS créent pour vous. Les ressources que vous utilisez, mais qui ne sont pas dans votre Compte AWS, telles que les Clés détenues par AWS, ne sont pas prises en compte dans le calcul de ces quotas.

Si vous avez atteint une limite de ressources, les demandes de création d'une ressource supplémentaire de ce type génèrent un message d'erreur `LimitExceededException`.

Tous les quotas de ressources AWS KMS sont ajustables, à l'exception du quota de la taille du document de politique de clés. Pour demander une augmentation de quota, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas. Pour demander une réduction de quota, pour modifier un quota qui ne figure pas dans la liste des Service Quotas, ou pour modifier un quota dans une Région AWS où les Service Quotas pour AWS KMS ne sont pas disponibles, rendez-vous dans le Centre AWS Support et créez un dossier.

Le tableau suivant répertorie et décrit les quotas de ressources AWS KMS dans chaque Compte AWS et région.

Nom du quota	Valeur par défaut	S'applique à	Ajustable
AWS KMS keys	100 000	Clés gérées par le client	Oui
Alias par clé KMS	50	Alias créés par le client	Oui
Octrois par clé KMS	50 000	Clés gérées par le client	Oui
Taille du document de stratégie de clé	32 Ko (32 768 octets)	Clés gérées par le client Clés gérées par AWS	Non
Quota de ressources du magasin de clé personnalisé	10	Compte AWS et région	Oui

En plus des quotas de ressources, AWS KMS utilise des quotas de demandes pour assurer la réactivité du service. Pour plus de détails, veuillez consulter the section called "Quotas de demande".

AWS KMS keys :100 000

Vous pouvez avoir jusqu'à 100 000 clés gérées par le client dans chaque région de votre Compte AWS. Ce quota s'applique à toutes les clés gérées par le client dans toutes les Régions AWSindépendamment de leur Spécification de clé ou état de clé. Chaque clé KMS est considérée comme une ressource unique. Les Clés gérées par AWS et Clés détenues par AWS ne sont pas prises en compte dans ce quota.

Alias par clé KMS : 50

Vous pouvez associer jusqu'à 50 alias avec chaque clé gérée par le client. Les alias qu'AWS associe aux Clés gérées par AWS ne sont pas pris en compte dans ce quota. Vous pouvez rencontrer ce quota lorsque vous créez ou mettez à jour un alias.

> **Note**
>
> La ResourceAliases condition kms : n'est effective que lorsque la clé KMS est conforme à ce quota. Si une clé KMS dépasse ce quota, les mandataires autorisés à utiliser la clé KMS par la condition `kms:ResourceAliases` se voient refuser l'accès à la clé KMS. Pour plus de détails, veuillez consulter Accès refusé en raison d'un quota d'alias.

Le quota d'alias par clé KMS remplace le quota d'alias par région qui limitait le nombre total d'alias dans chaque région d'un Compte AWS. AWS KMS a éliminé le quota d'alias par région.

Octrois par clé KMS : 50 000

Chaque clé gérée par le client peut avoir jusqu'à 50 000 octrois, y compris les octrois créés par les services AWS qui sont intégrés à AWS KMS. Ce quota ne s'applique pas aux Clés gérées par AWS ou Clés détenues par AWS.

L'un des effets de ce quota est que vous ne pouvez pas exécuter plus de 50 000 opérations autorisées par octroi qui utilisent simultanément la même clé KMS. Une fois que vous avez atteint le quota, vous pouvez créer de nouveaux octrois sur la clé KMS uniquement lorsqu'un octroi actif est retiré ou révoqué.

Par exemple, lorsque vous attachez un volume Amazon Elastic Block Store (Amazon EBS) à une instance Amazon Elastic Compute Cloud (Amazon EC2), le volume est déchiffré afin que vous puissiez le lire. Pour obtenir l'autorisation de déchiffrer les données, Amazon EBS crée un octroi pour chaque volume. Par conséquent, si tous vos volumes Amazon EBS utilisent la même clé KMS, vous ne pouvez pas attacher plus de 50 000 volumes en même temps.

Taille du document de stratégie de clé : 32 Ko

La longueur maximale de chaque document de stratégie de clé est 32 Ko (32 768 octets). Si vous utilisez un document de stratégie plus volumineux pour créer ou mettre à jour la stratégie d'une clé KMS, l'opération échoue.

Il ne s'agit pas d'un quota ajustable. Vous ne pouvez pas l'augmenter en utilisant Service Quotas ou en créant une requête dans AWS Support. Si votre stratégie de clé approche de la limite, envisagez d'utiliser les octrois plutôt que des instructions de stratégie. Les subventions sont particulièrement bien adaptées aux autorisations temporaires ou très spécifiques.

Vous utilisez un document de politique clé chaque fois que vous créez ou modifiez une politique clé en utilisant la vue ou la vue de stratégie par défaut dans l'AWS Management Consoleopération ou dans l'PutKeyPolicyopération. Cette limite s'applique à votre document de stratégie de clé, même si vous utilisez la vue par défaut dans la console AWS KMS, où vous ne modifiez pas directement les instructions JSON.

Quota de ressources des magasins de clés personnalisés : 10

Vous pouvez créer jusqu'à 10 magasins de clés personnalisés dans chaque Compte AWS et région. Si vous essayez d'en créer d'autres, l'CreateCustomKeyStoreopération échoue.

Ce quota s'applique au nombre total de magasins de clés personnalisés dans chaque compte et région, y compris tous les magasins de clés AWS CloudHSM et les magasins de clés externes, quel que soit leur état de connexion.

Quotas de demande

AWS KMS établit des quotas sur le nombre d'opérations d'API demandées par seconde. Les quotas de demande diffèrent en fonction de l'opération d'API, de la Région AWS et d'autres facteurs, tels que le type de clé KMS. Lorsque vous dépassez un quota de demande d'API, AWS KMS limite la demande.

Toutes les demandes de quotas AWS KMS sont ajustables, sauf pour le quota de demandes du magasin de clés AWS CloudHSM. Pour demander une augmentation de quota, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas. Pour demander une réduction de quota, pour modifier un quota qui ne figure pas dans la liste des Service Quotas, ou pour modifier un quota dans une Région AWS où les Service Quotas pour AWS KMS ne sont pas disponibles, rendez-vous dans le Centre AWS Support et créez un dossier.

Si vous dépassez le quota de demandes pour l'GenerateDataKeyopération, pensez à utiliser la fonctionnalité de mise en cache des clés de données duAWS Encryption SDK. La réutilisation de clés de données peut réduire la fréquence de vos demandes adressées à AWS KMS.

En plus des quotas de demande, AWS KMS utilise les quotas de ressources pour assurer la capacité de tous les utilisateurs. Pour plus de détails, veuillez consulter Quotas de ressources.

Pour afficher les tendances de vos taux de demande, utilisez la console Service Quotas. Vous pouvez également créer une CloudWatch alarme Amazon qui vous avertit lorsque le taux de demandes atteint un certain pourcentage de la valeur du quota. Pour plus de détails, consultez Gérer

vos taux de demandes AWS KMS d'API à l'aide de Service Quotas et d'Amazon CloudWatch dans le blog sur la AWS sécurité.

Rubriques

- Demander des quotas pour chaque opération d'API AWS KMS
- Application des quotas de demande
- Quotas partagés pour les opérations de chiffrement
- Demandes d'API effectuées en votre nom
- Demandes entre comptes
- Quotas de demandes de magasin de clés personnalisé

Demander des quotas pour chaque opération d'API AWS KMS

Ce tableau répertorie le code de quota des Service Quotas et la valeur par défaut de chaque quota de demande AWS KMS. Toutes les demandes de quotas AWS KMS sont ajustables, sauf pour le quota de demandes de magasin de clés AWS CloudHSM.

> **Note**
>
> Il peut être nécessaire de faire défiler horizontalement ou verticalement pour afficher toutes les données de ce tableau.

Nom du quota	Valeur par défaut (requêtes par seconde)

`Cryptographic operations (symmetric) request rate`
 	
S'applique à :
 	
• `Decrypt`
• `Encrypt`
• `GenerateDataKey`
• `GenerateDataKeyWithoutPlaintext`
	Ces quotas partagés varient selon la Région AWS et le type de clé KMS utilisé dans la demande. Chaque quota est calculé séparément.
	
	• 5 500 (partagées)
	• 10 000 (partagées) dans les régions suivantes :
	  • USA Est (Ohio), us-east-2
	  • Asie-Pacifique (Singapour), ap-southeast-1

Nom du quota	Valeur par défaut (requêtes par seconde)
• `GenerateMac` • `GenerateRandom` • `ReEncrypt` • `VerifyMac`	• Asie-Pacifique (Sydney), ap-southeast-2 • Asie-Pacifique (Tokyo), ap-northeast-1 • Europe (Francfort), eu-central-1 • Europe (Londres), eu-west-2 • 50 000 (partagées) dans les régions suivantes : • USA Est (Virginie du Nord), us-east-1 • USA Ouest (Oregon), us-west-2 • Europe (Irlande), eu-west-1
`Cryptographic operations (RSA) request rate` S'applique à : • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	500 (partagées) pour les clés KMS RSA
`Cryptographic operations (ECC) request rate` S'applique à : • `Sign` • `Verify`	300 (partagées) pour les clés KMS de courbe elliptique (ECC)

Nom du quota	Valeur par défaut (requêtes par seconde)
`Cryptographic operations (SM) request rate` S'applique à : • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	300 clés (partagées) KMS SM2 (régions de Chine uniquement)
`Custom key store request quotas` S'applique à : • `Decrypt` • `Encrypt` • `GenerateDataKey` • `GenerateDataKeyWithoutPlaintext` • `GenerateRandom` • `ReEncrypt`	Les quotas de demandes de magasin de clés personnalisé sont calculés séparément pour chaque magasin de clés personnalisé • 1 800 (partagées) pour chaque magasin de clés AWS CloudHSM • 1 800 (partagées) pour chaque magasin de clés externes
`CancelKeyDeletion request rate`	5
`ConnectCustomKeyStore request rate`	5
`CreateAlias request rate`	5
`CreateCustomKeyStore request rate`	5
`CreateGrant request rate`	50
`CreateKey request rate`	5

Nom du quota	Valeur par défaut (requêtes par seconde)
`DeleteAlias request rate`	15
`DeleteCustomKeyStore request rate`	5
`DeleteImportedKeyMaterial request rate`	5
`DescribeCustomKeyStores request rate`	5
`DescribeKey request rate`	2000
`DisableKey request rate`	5
`DisableKeyRotation request rate`	5
`DisconnectCustomKeyStore request rate`	5
`EnableKey request rate`	5
`EnableKeyRotation request rate`	15
`GenerateDataKeyPair (ECC_NIST_P256) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100

Nom du quota	Valeur par défaut (requêtes par seconde)
`GenerateDataKeyPair (ECC_NIST_P384) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (ECC_NIST_P521) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (ECC_SECG_P256K1) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (RSA_2048) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	1

Nom du quota	Valeur par défaut (requêtes par seconde)
`GenerateDataKeyPair (RSA_3072) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,5 (1 dans chaque intervalle de 2 secondes)
`GenerateDataKeyPair (RSA_4096) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,1 (1 dans chaque intervalle de 10 secondes)
`GenerateDataKeyPair (SM2 – China Regions only) request rate` S'applique à : • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	25
`GetKeyPolicy request rate`	1 000
`GetKeyRotationStatus request rate`	1 000
`GetParametersForImport request rate`	0,25 (1 dans chaque intervalle de 4 secondes)
`GetPublicKey request rate`	2000
`ImportKeyMaterial request rate`	5

Nom du quota	Valeur par défaut (requêtes par seconde)
`ListAliases request rate`	500
`ListGrants request rate`	100
`ListKeyPolicies request rate`	100
`ListKeys request rate`	500
`ListResourceTags request rate`	2000
`ListRetirableGrants request rate`	100
`PutKeyPolicy request rate`	15
`ReplicateKey request rate` Une opération `ReplicateKey` compte comme une demande `ReplicateKey` dans la région de la clé principale et deux demandes `CreateKey` dans la région du réplica. L'une des demandes `CreateKey` est un essai pour détecter les problèmes potentiels avant de créer la clé.	5
`RetireGrant request rate`	30
`RevokeGrant request rate`	30
`ScheduleKeyDeletion request rate`	15
`TagResource request rate`	10
`UntagResource request rate`	5
`UpdateAlias request rate`	5
`UpdateCustomKeyStore request rate`	5
`UpdateKeyDescription request rate`	5

Nom du quota	Valeur par défaut (requêtes par seconde)
`UpdatePrimaryRegion request rate` Une opération `UpdatePrimaryRegion` compte comme deux demandes `UpdatePrimaryRegion` ; une demande dans chacune des deux régions touchées.	5

Application des quotas de demande

Lors de la révision des quotas de demande, gardez à l'esprit les informations suivantes.

- Les quotas de demande s'appliquent aux clés gérées par le client et aux Clés gérées par AWS. L'utilisation des Clés détenues par AWS n'est pas prise en compte dans les quotas de demande de votre Compte AWS, même lorsqu'elles sont utilisées pour protéger les ressources de votre compte.
- Les quotas de demande s'appliquent aux demandes envoyées aux points de terminaison FIPS et aux points de terminaison non FIPS. Pour obtenir la liste complète des points de terminaison AWS KMS, consultez Points de terminaison et quotas de service AWS Key Management Service dans le Références générales AWS.
- La limitation est basée sur toutes les demandes concernant les clés KMS de tous types dans la région. Ce total comprend les demandes provenant de tous les principaux du Compte AWS, y compris les demandes de services AWS en votre nom.
- Chaque quota de demande est calculé indépendamment. Par exemple, les demandes relatives à l'CreateKeyopération n'ont aucun effet sur le quota de demandes pour l'CreateAliasopération. Si vos demandes `CreateAlias` sont soumises à une limitation, vos demandes `CreateKey` peuvent tout de même s'exécuter avec succès.
- Bien que les opérations de chiffrement partagent un quota, le quota partagé est calculé indépendamment des quotas appliqués aux autres opérations. Par exemple, les appels aux opérations de chiffrement et de déchiffrement partagent un quota de demandes, mais ce quota est indépendant du quota pour les opérations de gestion, telles que. EnableKey Par exemple, dans la région Europe (Londres), vous pouvez effectuer 10 000 opérations de chiffrement sur des clés KMS symétriques plus 5 opérations `EnableKey` par seconde sans être limité.

Quotas partagés pour les opérations de chiffrement

Les opérations de chiffrement AWS KMS partagent des quotas de demandes. Vous pouvez demander n'importe quelle combinaison d'opérations de chiffrement prises en charge par la clé KMS, à condition que le nombre total d'opérations de chiffrement ne dépasse pas le quota de demande pour ce type de clé KMS. Les exceptions sont GenerateDataKeyPairet GenerateDataKeyPairWithoutPlaintext, qui partagent un quota distinct.

Les quotas des différents types de clés KMS sont calculés indépendamment. Chaque quota s'applique à toutes les demandes concernant ces opérations dans le Compte AWS et la région avec le type de clé donné dans chaque intervalle d'une seconde.

- Le taux de demande des opérations de chiffrement (symétrique) est le quota de demande partagé pour les opérations de chiffrement utilisant des clés KMS symétriques dans un compte et une région. Ce quota s'applique aux opérations cryptographiques avec des clés de chiffrement symétriques et des clés HMAC, qui sont également symétriques.

  Par exemple, vous pouvez utiliser des clés KMS symétriques dans une Région AWS avec un quota partagé de 10 000 demandes par seconde. Lorsque vous faites 7 000 GenerateDataKeydemandes par seconde et 2 000 demandes de déchiffrement par seconde, AWS KMS cela ne limite pas vos demandes. Par contre, si vous effectuez 9 500 demandes `GenerateDataKey` et 1 000 demandes Encrypt par seconde, AWS KMS limite vos demandes, car elles dépassent le quota partagé.

  Les opérations de chiffrement sur les clés KMS de chiffrement symétrique d'un magasin de clés personnalisé sont à la fois comptabilisées dans le taux de demandes d'opérations de chiffrement (symétrique) du compte et dans le quota de demandes du magasin de clés personnalisé.
- Le taux de demande RSA (opérations de chiffrement) est le quota de demande partagé pour les opérations de chiffrement utilisant des clés KMS asymétriques RSA.

  Par exemple, avec un quota de demandes de 500 opérations par seconde, vous pouvez effectuer 200 demandes Encrypt et 100 demandes Decrypt avec des clés KMS RSA capables de chiffrer et de déchiffrer, plus 50 demandes Sign et 150 demandes Verify avec des clés KMS RSA qui peuvent signer et vérifier.
- Le taux de demande d'opérations de chiffrement (ECC) est le quota de demande partagé pour les opérations de chiffrement utilisant des clés KMS asymétriques de courbe elliptique (ECC).

  Par exemple, avec un quota de demandes de 300 opérations par seconde, vous pouvez effectuer 100 demandes Sign et 200 demandes Verify avec des clés KMS RSA qui peuvent signer et vérifier.

- Le taux de requêtes d'opérations cryptographiques (SM — Régions de Chine seulement) est le quota de requêtes partagé pour les opérations cryptographiques utilisant des clés SM KMS asymétriques.

  Par exemple, avec un quota de demandes de 300 opérations par seconde, vous pouvez effectuer 100 demandes Encrypt (Chiffrées) et 100 demandes Decrypt (Déchiffrées) avec des clés KMS SM2 capables de chiffrer et de déchiffrer, plus 50 demandes Sign (Signer) et 50 demandes Verify (Vérifier) avec des clés KMS SM2 qui peuvent signer et vérifier.
- Le quota de demandes du magasin de clés personnalisé désigne le quota de demandes partagées pour les opérations de chiffrement sur les clés KMS d'un magasin de clés personnalisé. Cette limite est calculée séparément pour chaque magasin de clés personnalisé.

  Les opérations de chiffrement sur les clés KMS de chiffrement symétrique d'un magasin de clés personnalisé sont à la fois comptabilisées dans le taux de demandes d'opérations de chiffrement (symétrique) du compte et dans le quota de demandes du magasin de clés personnalisé.

Les quotas des différents types de clés sont également calculées indépendamment. Par exemple, dans la région Asie-Pacifique (Singapour), si vous utilisez à la fois des clés KMS symétriques et asymétriques, vous pouvez effectuer jusqu'à 10 000 appels par seconde avec des clés KMS symétriques (y compris des clés HMAC) plus 500 appels supplémentaires par seconde avec vos clés KMS asymétriques RSA, plus 300 demandes supplémentaires par seconde avec vos clés KMS ECC.

Demandes d'API effectuées en votre nom

Vous pouvez effectuer des demandes d'API directement ou en utilisant un service AWS intégré qui effectue des demandes d'API à AWS KMS en votre nom. Le quota s'applique aux deux types de demandes.

Par exemple, vous pouvez stocker des données dans Simple Storage Service (Amazon S3) à l'aide du chiffrement côté serveur avec une clé KMS (SSE-KMS). Chaque fois que vous chargez ou téléchargez un objet S3 qui est chiffré avec SSE-KMS, Amazon S3 effectue une demande `GenerateDataKey` (pour les chargements) ou `Decrypt` (pour les téléchargements) à AWS KMS en votre nom. Ces demandes sont prises en compte dans le quota qui vous est imposé. C'est pourquoi AWS KMS limite les demandes si vous dépassez un total combiné de 5 500 (ou 10 000 ou 50 000 selon votre Région AWS) chargements ou téléchargements par seconde d'objets S3 chiffrés avec SSE-KMS.

Demandes entre comptes

Lorsqu'une application dans un Compte AWS utilise une clé KMS appartenant à un autre compte, cette action est qualifiée de demande inter-comptes. Pour les demandes inter-comptes, AWS KMS limite le compte qui effectue les demandes, et non pas le compte qui possède la clé KMS. Par exemple, si une application du compte A utilise une clé KMS du compte B, l'utilisation de la clé KMS est uniquement soumise aux quotas du compte A.

Quotas de demandes de magasin de clés personnalisé

AWS KMS gère les quotas de demandes pour les opérations de chiffrement sur les clés KMS d'un magasin de clés personnalisé. Ces quotas de demandes sont calculés séparément pour chaque magasin de clés personnalisé.

Quota de requêtes de magasin de clés personnalisé	Valeur par défaut (demandes par seconde) pour chaque magasin de clés personnalisé	Ajustable
Quota de demandes de magasin de clés AWS CloudHSM	1800	Non
Quota de demandes de magasin de clés externes	1800	Oui

> **Note**
>
> Le quota de demandes de magasin de clés personnalisé AWS KMS n'apparaît pas dans la console Service Quotas. Vous ne pouvez ni consulter, ni gérer ces quotas à l'aide des opérations de l'API Service Quotas. Pour solliciter une modification de votre quota de demandes de magasin de clés externes, accédez au Centre AWS Support et créez une demande.
>
> Si le AWS CloudHSM cluster associé à un magasin de AWS CloudHSM clés traite de nombreuses commandes, y compris celles qui ne sont pas liées au magasin de clés personnalisé, vous pourriez obtenir un AWS KMS `ThrottlingException` lower-than-expected at. Si cela se produit, réduisez votre taux de requêtes adressées à AWS KMS,

réduisez la charge non associée ou utilisez un cluster AWS CloudHSM dédié pour votre magasin de clés AWS CloudHSM.
AWS KMSsignale la limitation des demandes de stockage de clés externes dans la `ExternalKeyStoreThrottle` CloudWatch métrique. Vous pouvez utiliser cette métrique pour visualiser les modèles de limitation, créer des alertes et ajuster votre quota de demandes pour le magasin de clés externes.

Une demande d'opération de chiffrement sur une clé KMS d'un magasin de clés personnalisé compte pour deux quotas :

- Quota de taux de demandes d'opérations de chiffrement (symétrique) (par compte)

  Les demandes d'opérations de chiffrement sur les clés KMS d'un magasin de clés personnalisé sont comptabilisées dans le quota `Cryptographic operations (symmetric) request rate` de chaque région Compte AWS. Par exemple, dans la région USA Est (Virginie du Nord) (us-east-1), chaque Compte AWS peut recevoir jusqu'à 50 000 demandes par seconde sur des clés KMS de chiffrement symétrique, y compris des demandes utilisant une clé KMS figurant dans un magasin de clés personnalisé.
- Quota de demandes de magasin de clés personnalisé (par magasin de clés personnalisé)

  Les demandes d'opérations de chiffrement sur les clés KMS d'un magasin de clés personnalisé sont également comptabilisées dans le `Custom key store request quota` de 1 800 opérations par seconde. Ces quotas sont calculés séparément pour chaque magasin de clés personnalisé. Ils peuvent inclure des demandes provenant de plusieurs Comptes AWS qui utilisent des clés KMS du magasin de clés personnalisé.

Par exemple, une opération Encrypt (Chiffrer) sur une clé KMS d'un magasin de clés personnalisé (quel que soit son type) dans la région USA Est (Virginie du Nord) (us-east-1) est comptabilisée dans le quota au niveau du compte du `Cryptographic operations (symmetric) request rate` (50 000 demandes par seconde) pour son compte et sa région, et dans un `Custom key store request quota` (1 800 demandes par seconde) pour son magasin de clés personnalisé. Toutefois, une demande d'opération de gestion, telle que celle portant sur une clé KMS dans un magasin de clés personnalisé PutKeyPolicy, ne s'applique qu'au quota au niveau du compte (15 demandes par seconde).

Limitation des demandes AWS KMS

Pour s'assurer que AWS KMS peut fournir des réponses rapides et fiables aux demandes d'API des clients, le service limite les demandes d'API qui excédent certaines limites.

La limitation se produit lorsque AWS KMS rejette une demande qui pourrait autrement être valide, et renvoie une erreur `ThrottlingException` comme la suivante.

```
You have exceeded the rate at which you may call KMS. Reduce the frequency of your
 calls.
(Service: AWSKMS; Status Code: 400; Error Code: ThrottlingException; Request ID: <ID>
```

AWS KMS limite les demandes pour les conditions suivantes.

- Le taux de demande par seconde dépasse le quota de demandes AWS KMS pour un compte et une région.

  Par exemple, si les utilisateurs de votre compte soumettent 1 000 demandes `DescribeKey` en une seconde, AWS KMS limite toutes les demandes `DescribeKey` ultérieures à cette seconde.

  Pour répondre à la limitation, utilisez une stratégie d'interruption et de nouvelle tentative. Cette stratégie est implémentée automatiquement pour les erreurs HTTP 400 dans certains kits SDK AWS.
- Une salve ou un taux élevé soutenu de demandes de modification de l'état de la même clé KMS. Cette condition est souvent connue sous le nom de « touche de raccourci ».

  Par exemple, si une application de votre compte envoie une salve persistante de demandes `EnableKey` et `DisableKey` pour la même clé KMS, AWS KMS limite les demandes. Cette limitation se produit même si les demandes ne dépassent pas la limite de request-per-second demandes pour les opérations `EnableKey` et`DisableKey`.

  Pour répondre à la limitation, ajustez votre la logique d'application, afin qu'elle ne fasse que les demandes requises ou qu'elle consolide les demandes de plusieurs fonctions.
- Les demandes d'opérations sur les clés KMS dans un magasin de AWS CloudHSM clés peuvent être limitées à un lower-than-expected rythme tel que le AWS CloudHSM cluster associé au magasin de AWS CloudHSM clés traite de nombreuses commandes, y compris celles qui ne sont pas liées au magasin de AWS CloudHSM clés.

(AWS KMS ne limite plus les requêtes d'opérations sur les clés KMS dans un magasin de clés AWS CloudHSM lorsqu'il n'y a pas de sessions PKCS #11 disponibles pour le cluster AWS CloudHSM. Au lieu de cela, il lève une exception `KMSInternalException` et vous recommande de relancer votre requête.)

Pour afficher les tendances de vos taux de demande, utilisez la console Service Quotas. Vous pouvez également créer une CloudWatch alarme Amazon qui vous avertit lorsque le taux de demandes atteint un certain pourcentage de la valeur du quota. Pour plus de détails, consultez Gérer vos taux de demandes AWS KMS d'API à l'aide de Service Quotas et d'Amazon CloudWatch dans le blog sur la AWS sécurité.

Tous les quotas AWS KMS sont ajustables, à l'exception du quota de ressources de la taille du document de politique de clé et du AWS CloudHSMquota de ressources des magasins de clés personnalisés. Pour demander une augmentation de quota, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas. Pour demander une réduction de quota, pour modifier un quota qui ne figure pas dans la liste des Service Quotas, ou pour modifier un quota dans une Région AWS où les Service Quotas pour AWS KMS ne sont pas disponibles, rendez-vous dans le Centre AWS Support et créez un dossier.

> **Note**
>
> Le quota de demandes de magasin de clés personnalisé AWS KMS n'apparaît pas dans la console Service Quotas. Vous ne pouvez ni consulter, ni gérer ces quotas à l'aide des opérations de l'API Service Quotas. Pour solliciter une modification de votre quota de demandes de magasin de clés externes, accédez au Centre AWS Support et créez une demande.

Comment les services AWS utilisent AWS KMS

De nombreux services AWS utilisent AWS KMS pour prendre en charge le chiffrement de vos données. Lorsqu'un service AWS est intégré à AWS KMS, vous pouvez utiliser les AWS KMS keys dans votre compte pour protéger les données que le service reçoit, stocke ou gère pour vous. Pour obtenir la liste complète des services AWS intégrés à AWS KMS, veuillez consulter Intégration des services AWS.

Les rubriques suivantes présentent en détail la manière dont les services particuliers utilisent AWS KMS, y compris les clés KMS qu'ils prennent en charge, la manière dont ils gèrent les clés de données, les autorisations dont ils ont besoin et la manière de suivre l'utilisation des clés KMS par chaque service dans votre compte.

> **Important**
>
> Les services AWS qui sont intégrés à AWS KMS utilisent des clés KMS de chiffrement symétriques pour chiffrer vos données. Ces services ne prennent pas en charge le chiffrement avec des clés KMS asymétriques. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Rubriques

- Comment AWS CloudTrail utilise AWS KMS
- Comment Amazon DynamoDB utilise AWS KMS
- Comment Amazon Elastic Block Store (Amazon EBS) utilise AWS KMS
- Comment Amazon Elastic Transcoder utilise AWS KMS
- Comment Amazon EMR utilise AWS KMS
- Comment AWS Nitro Enclaves utilise AWS KMS
- Comment d'Amazon Redshift utilise AWS KMS
- Comment Amazon Relational Database Service (Amazon RDS) utilise AWS KMS
- Comment AWS Secrets Manager utilise AWS KMS
- Comment Amazon Simple Email Service (Amazon SES) utilise AWS KMS
- Comment Amazon Simple Storage Service (Amazon S3) utilise AWS KMS

- Comment AWS Systems Manager Parameter Store utilise AWS KMS
- Comment Amazon WorkMail utilise AWS KMS
- Comment WorkSpaces utilise AWS KMS

Comment AWS CloudTrail utilise AWS KMS

Vous pouvez utiliser AWS CloudTrail pour enregistrer les appels d'API AWS et d'autres activités pour votre Compte AWS et pour enregistrer les informations enregistrées dans des fichiers journaux, dans un compartiment Amazon Simple Storage Service (Amazon S3) de votre choix. Par défaut, les fichiers journaux placés dans votre compartiment S3 sont chiffrés à l'aide du chiffrement côté serveur avec des clés de chiffrement gérées par Amazon S3 (SSE-S3). CloudTrail Mais vous pouvez choisir à la place d'utiliser un chiffrement côté serveur avec une clé gérée KMS (SSE-KMS). Pour savoir comment chiffrer vos fichiers CloudTrail journaux avecAWS KMS, consultez la section Chiffrement des fichiers CloudTrail journaux avec AWS KMS keys (SSE-KMS) dans le guide de l'utilisateur. AWS CloudTrail

> **Important**
>
> AWS CloudTrail et Amazon S3 prennent uniquement en charge les AWS KMS keys symétriques. Vous ne pouvez pas utiliser de clé KMS asymétrique pour chiffrer vos CloudTrail journaux. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, consultez Identification des clés KMS asymétriques.

Vous ne payez pas de frais d'utilisation des clés lorsque vous lisez CloudTrail ou écrivez des fichiers journaux chiffrés avec une clé SSE-KMS. Toutefois, vous payez des frais d'utilisation des clés lorsque vous accédez à des fichiers CloudTrail journaux chiffrés avec une clé SSE-KMS. Pour plus d'informations sur la tarification AWS KMS, consultez Tarification AWS Key Management Service. Pour plus d'informations sur la CloudTrail tarification, consultez AWS CloudTrailles sections Tarification et Gestion des coûts dans le Guide de AWS CloudTrail l'utilisateur.

Rubriques

- Comprendre quand votre clé KMS est utilisée

Comprendre quand votre clé KMS est utilisée

Chiffrement des fichiers CloudTrail journaux à l'aide de AWS KMS versions basées sur la fonctionnalité Amazon S3 appelée chiffrement côté serveur avec un AWS KMS key (SSE-KMS). Pour en savoir plus sur SSE-KMS, veuillez consulter Comment Amazon Simple Storage Service (Amazon S3) utilise AWS KMS dans ce guide ou Protection des données grâce au chiffrement côté serveur avec des clés KMS (SSE-KMS) dans le guide du développeur Amazon Simple Storage Service.

Lorsque vous configurez AWS CloudTrail pour utiliser SSE-KMS pour chiffrer vos fichiers journaux, Amazon CloudTrail S3 utilise les vôtres AWS KMS keys lorsque vous effectuez certaines actions avec ces services. Les sections suivantes expliquent quand et comment ces services peuvent utiliser votre clé KMS, et fournissent des informations supplémentaires que vous pouvez utiliser pour valider cette explication.

Actions entraînant l'utilisation CloudTrail de votre clé KMS par Amazon S3

- Vous configurez CloudTrail pour crypter les fichiers journaux avec votre AWS KMS key
- CloudTrail place un fichier journal dans votre compartiment S3
- Vous obtenez un fichier journal chiffré à partir de votre compartiment S3

Vous configurez CloudTrail pour crypter les fichiers journaux avec votre AWS KMS key

Lorsque vous mettez à jour votre CloudTrail configuration pour utiliser votre clé KMS, CloudTrail envoie une `GenerateDataKey`demande AWS KMS pour vérifier que la clé KMS existe et que CloudTrail vous êtes autorisé à l'utiliser pour le chiffrement. CloudTrail n'utilise pas la clé de données obtenue.

La demande `GenerateDataKey` inclut les informations suivantes pour le contexte de chiffrement :

- Le nom de ressource Amazon (ARN) du CloudTrail parcours
- L'ARN du compartiment S3 et le chemin où les fichiers CloudTrail journaux sont livrés

La `GenerateDataKey` demande entraîne une entrée dans vos CloudTrail journaux, comme dans l'exemple suivant. Lorsque vous voyez une entrée de journal comme celle-ci, vous pouvez déterminer que CloudTrail (1) a appelé l'AWS KMS`GenerateDataKey`opération (3)

pour un suivi spécifique
(4).
2
AWS KMSa créé la clé de données sous une clé KMS spécifique
(5).

> **Note**
>
> Vous devrez peut-être faire défiler l'affichage vers la droite pour voir certaines des alertes dans l'exemple suivant d'entrée de journal.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::086441151436:user/
AWSCloudTrail",1
    "accountId": "086441151436",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "AWSCloudTrail",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2015-11-11T21:15:33Z"
    }},
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:15:33Z",
  "eventSource":
 "kms.amazonaws.com",2
  "eventName":
 "GenerateDataKey",3
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAliasForCloudTrailKMS
 key",
    "encryptionContext": {
```

```
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",4
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/AWSLogs/111122223333/"
    },
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "581f1f11-88b9-11e5-9c9c-595a1fb59ac0",
  "eventID": "3cdb2457-c035-4890-93b6-181832b9e766",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",5
    "accountId": "111122223333"
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
}
```

CloudTrail place un fichier journal dans votre compartiment S3

Chaque fois CloudTrail qu'un fichier journal est placé dans votre compartiment S3, Amazon S3 envoie une `GenerateDataKey`demande AWS KMS au nom de CloudTrail. En réponse à cette demande, AWS KMS génère une clé de données unique, puis envoie à Amazon S3 deux copies de la clé de données : l'une en texte brut et l'autre chiffrée avec la clé KMS spécifiée. Amazon S3 utilise la clé de données en texte brut pour chiffrer le fichier CloudTrail journal, puis la supprime de la mémoire dès que possible après utilisation. Amazon S3 stocke la clé de données chiffrée sous forme de métadonnées dans le fichier CloudTrail journal chiffré.

La demande `GenerateDataKey` inclut les informations suivantes pour le contexte de chiffrement :

- Le nom de ressource Amazon (ARN) du CloudTrail parcours
- L'ARN de l'objet S3 (le fichier CloudTrail journal)

Chaque `GenerateDataKey` demande entraîne une entrée dans vos CloudTrail journaux, comme dans l'exemple suivant. Lorsque vous voyez une entrée de journal comme celle-ci, vous pouvez déterminer que CloudTrail (1) a appelé l'AWS KMS`GenerateDataKey`opération

(3))
pour un suivi spécifique
(4))
afin de protéger un fichier journal spécifique
(5)).
2
AWS KMSa créé la clé de données sous la clé KMS spécifiée
(6)),
affichée deux fois dans la même entrée de journal.

> **Note**
>
> Vous devrez peut-être faire défiler l'affichage vers la droite pour voir certaines des alertes dans l'exemple suivant d'entrée de journal.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:i-34755b85",
    "arn": "arn:aws:sts::086441151436:assumed-role/AWSCloudTrail/
i-34755b85",1
    "accountId": "086441151436",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-11-11T20:45:25Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::086441151436:role/AWSCloudTrail",
        "accountId": "086441151436",
        "userName": "AWSCloudTrail"
      }
    },
    "invokedBy": "internal.amazonaws.com"
```

```
  },
  "eventTime": "2015-11-11T21:15:58Z",
  "eventSource":
 "kms.amazonaws.com",(2)
  "eventName":
 "GenerateDataKey",(3)
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",(4)
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/
AWSLogs/111122223333/CloudTrail/us-west-2/2015/11/11/111122223333_CloudTrail_us-
west-2_20151111T2115Z_7JREEBimdK8d2nC9.json.gz"(5)
    },
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",(6)
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "66f3f74a-88b9-11e5-b7fb-63d925c72ffe",
  "eventID": "7738554f-92ab-4e27-83e3-03354b1aa898",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",(6)
    "accountId": "111122223333"
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
}
```

Vous obtenez un fichier journal chiffré à partir de votre compartiment S3

Chaque fois que vous recevez un fichier CloudTrail journal chiffré depuis votre compartiment S3, Amazon S3 envoie une `Decrypt`demande à en votre AWS KMS nom pour déchiffrer la clé de données chiffrée du fichier journal. En réponse à cette demande, AWS KMS utilise votre clé KMS pour déchiffrer la clé de données, puis envoie la clé de données en texte brut à Amazon S3. Amazon

S3 utilise la clé de données en texte brut pour déchiffrer le fichier CloudTrail journal, puis la supprime de la mémoire dès que possible après utilisation.

La demande `Decrypt` inclut les informations suivantes pour le contexte de chiffrement :

- Le nom de ressource Amazon (ARN) du CloudTrail parcours
- L'ARN de l'objet S3 (le fichier CloudTrail journal)

Chaque `Decrypt` demande entraîne une entrée dans vos CloudTrail journaux, comme dans l'exemple suivant. Lorsque vous voyez une entrée de journal comme celle-ci, vous pouvez déterminer qu'un utilisateur de votre Compte AWS (1) a appelé l'opération AWS KMS (2) `Decrypt` (3) pour obtenir un journal de suivi spécifique (4) et un fichier journal spécifique (5). AWS KMS a déchiffré la clé de données sous une clé KMS spécifique (6).

> **Note**
>
> Vous devrez peut-être faire défiler l'affichage vers la droite pour voir certaines des alertes dans l'exemple suivant d'entrée de journal.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:role/cloudtrail-
admin",(1)
    "accountId": "111122223333",
```

```
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "cloudtrail-admin",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2015-11-11T20:48:04Z"
    }},
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:20:52Z",
  "eventSource":
 "kms.amazonaws.com",(2)
  "eventName":
 "Decrypt",(3)
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",(4)
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/
AWSLogs/111122223333/CloudTrail/us-west-2/2015/11/11/111122223333_CloudTrail_us-
west-2_20151111T2115Z_7JREEBimdK8d2nC9.json.gz"(5)
    }
  },
  "responseElements": null,
  "requestID": "16a0590a-88ba-11e5-b406-436f15c3ac01",
  "eventID": "9525bee7-5145-42b0-bed5-ab7196a16daa",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",(6)
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Comment Amazon DynamoDB utilise AWS KMS

Amazon DynamoDB est un service de base de données NoSQL évolutif et entièrement géré. DynamoDB s'intègre à AWS Key Management Service (AWS KMS) pour prendre en charge la fonction de chiffrement au repos côté serveur.

Avec le chiffrement au repos, DynamoDB chiffre de manière transparente toutes les données client dans une table DynamoDB, y compris sa clé principale et les index secondaires globaux et locaux, chaque fois que la table est stockée durablement sur disque. (Si votre table a une clé de tri, certaines clés de tri qui marquent les limites de plage sont stockées en texte brut dans les métadonnées de la table.) Lorsque vous accédez à votre table, DynamoDB déchiffre les données de table de manière transparente. Vous n'avez pas besoin de modifier vos applications pour utiliser ou gérer des tables chiffrées.

Le chiffrement au repos protège également les DynamoDB Streams, les tables globales et les sauvegardes chaque fois que ces objets sont enregistrés sur un support durable. Les instructions sur les tables de cette rubrique s'appliquent aussi à ces objets.

Toutes les tables DynamoDB sont chiffrées. Il n'y a aucune option pour activer ou désactiver le chiffrement pour les tables nouvelles ou existantes. Par défaut, toutes les tables sont chiffrées sous une Clé détenue par AWS dans le compte de service DynamoDB. Cependant, vous pouvez sélectionner une option pour chiffrer tout ou une partie de vos tables sous une clé gérée par le client ou avec la Clé gérée par AWS pour DynamoDB dans votre compte.

Pour en savoir plus sur la prise en charge des clés KMS par Amazon DynamoDB, veuillez consulter la rubrique Chiffrement de DynamoDB au repos dans le Guide du développeur Amazon DynamoDB.

Comment Amazon Elastic Block Store (Amazon EBS) utilise AWS KMS

Cette rubrique explique en détail comment Amazon Elastic Block Store (Amazon EBS) utilise AWS KMS pour chiffrer les volumes et les instantanés. Pour obtenir des instructions de base sur le chiffrement de volumes Amazon EBS, veuillez consulter Chiffrement Amazon EBS.

Rubriques

- Chiffrement Amazon EBS
- Utilisation des clés KMS et des clés de données
- Contexte du chiffrement Amazon EBS

- Détection des défaillances Amazon EBS
- Utilisation de AWS CloudFormation pour créer des volumes Amazon EBS chiffrés

Chiffrement Amazon EBS

Lorsque vous attachez un volume Amazon EBS chiffré à un type d'instance Amazon Elastic Compute Cloud (Amazon EC2) pris en charge, les données stockées au repos sur le volume, les I/O de disque et les instantanés créés à partir du volume sont tous chiffrés. Le chiffrement intervient sur les serveurs qui hébergent les instances Amazon EC2.

Cette fonction est prise en charge sur tous les types de volume Amazon EBS. Vous pouvez accéder aux volumes chiffrés de la même façon que vous accédez aux autres volumes. Le chiffrement et le déchiffrement sont gérés de façon transparente et ne nécessitent aucune action supplémentaire de votre part, de votre instance EC2 ou de votre application. Les instantanés de volumes chiffrés sont automatiquement chiffrés et les volumes créés à partir d'instantanés chiffrés sont également automatiquement chiffrés.

Le statut de chiffrement d'un volume EBS est déterminé lorsque vous créez le volume. Vous ne pouvez pas modifier le statut de chiffrement d'un volume existant. Par contre, vous pouvez migrer les données entre des volumes chiffrés et des volumes non chiffrés, et appliquer un nouveau statut de chiffrement lors de la copie d'un instantané.

Amazon EBS prend en charge le chiffrement facultatif par défaut. Vous pouvez activer automatiquement le chiffrement sur tous les nouveaux volumes EBS et les copies d'instantanés dans votre Compte AWS et votre région. Ce paramètre de configuration n'affecte pas les volumes ou instantanés existants. Pour plus d'informations, veuillez consulter Chiffrement par défaut dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux ou le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

Utilisation des clés KMS et des clés de données

Lorsque vous créez un volume Amazon EBS., vous spécifiez un AWS KMS key. Par défaut, Amazon EBS utilise la Clé gérée par AWS pour Amazon EBS dans votre compte (`aws/ebs`). Toutefois, vous pouvez spécifier une clé gérée par le client que vous créez et gérez.

Pour utiliser une clé gérée par le client, vous devez autoriser Amazon EBS à utiliser la clé KMS en votre nom. Pour obtenir la liste des autorisations requises, veuillez consulter Autorisations pour les utilisateurs IAM dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux ou le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

> **Important**
>
> Amazon EBS prend uniquement en charge les clés KMS symétriques. Vous ne pouvez pas utiliser une clé KMS asymétrique pour chiffrer un volume Amazon EBS. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Pour chaque volume, Amazon EBS demande à AWS KMS de générer une clé de données unique chiffrée sous la clé KMS que vous spécifiez. Amazon EBS stocke la clé de données chiffrée avec le volume. Ensuite, lorsque vous attachez le volume à une instance Amazon EC2, Amazon EBS appelle AWS KMS pour déchiffrer la clé de données. Amazon EBS utilise la clé de données en texte brut dans la mémoire de l'hyperviseur pour chiffrer toutes les I/O de disque sur le volume. Pour plus d'informations, veuillez consulter Fonctionnement du chiffrement EBS dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux ou le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

Contexte du chiffrement Amazon EBS

Dans ses demandes GenerateDataKeyWithoutPlaintextet Decrypt àAWS KMS, Amazon EBS utilise un contexte de chiffrement avec une paire nom-valeur qui identifie le volume ou le snapshot inclus dans la demande. Le nom du contexte de chiffrement ne varie pas.

Un contexte de chiffrement est un ensemble de paires clé-valeur qui contiennent des données non secrètes arbitraires. Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de données, AWS KMS lie de manière chiffrée le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez transmettre le même contexte de chiffrement.

Pour tous les volumes et pour les instantanés chiffrés créés avec l'CreateSnapshotopération Amazon EBS, Amazon EBS utilise l'ID du volume comme valeur de contexte de chiffrement. Dans le champ `requestParameters` d'une entrée de journal CloudTrail, le contexte de chiffrement ressemble à ce qui suit :

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

Pour les instantanés chiffrés créés avec l'opération Amazon CopySnapshotEC2, Amazon EBS utilise l'ID du snapshot comme valeur de contexte de chiffrement. Dans le champ `requestParameters` d'une entrée de journal CloudTrail, le contexte de chiffrement ressemble à ce qui suit :

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

Détection des défaillances Amazon EBS

Pour créer un volume EBS chiffré ou attacher le volume à une instance EC2, Amazon EBS et l'infrastructure Amazon EC2 doivent être en mesure d'utiliser la clé KMS que vous avez spécifiée pour le chiffrement des volumes EBS. Lorsque la clé KMS n'est pas utilisable, par exemple lorsque son état de clé n'est pas `Enabled`, la création ou l'attachement du volume échoue.

Dans ce cas, Amazon EBS envoie un événement à Amazon EventBridge (anciennement CloudWatch Events) pour vous informer de l'échec. Dans EventBridge, vous pouvez établir des règles qui déclenchent des actions automatiques en réponse à ces événements. Pour plus d'informations, consultez Amazon CloudWatch Events pour Amazon EBS dans le guide de l'utilisateur Amazon EC2 pour les instances Linux, en particulier les sections suivantes :

- Clé de chiffrement non valide pour l'attachement ou le réattachement du volume
- Clé de chiffrement non valide pour la création du volume

Pour résoudre ces problèmes, veillez à ce que la clé KMS spécifiée pour le chiffrement des volumes EBS soit activée. Pour cela, commencez par vérifier la clé KMS afin d'identifier son état actuel (la colonne Statut dans AWS Management Console). Ensuite, consultez les informations à l'aide de l'un des liens suivants :

- Si la clé KMS est désactivée, activez-la.
- Si la clé KMS est en attente d'importation, importez-la.
- Si la clé KMS est en attente de suppression, annulez cette suppression.

Utilisation de AWS CloudFormation pour créer des volumes Amazon EBS chiffrés

Vous pouvez utiliser AWS CloudFormation pour créer des volumes Amazon EBS chiffrés. Pour plus d'informations, consultez AWS::EC2::Volume dans le Guide de l'utilisateur AWS CloudFormation.

Comment Amazon Elastic Transcoder utilise AWS KMS

Vous pouvez utiliser Amazon Elastic Transcoder pour convertir des fichiers multimédias stockés dans un compartiment Amazon S3 vers des formats requis par les appareils de lecture grand public. Les fichiers d'entrée et de sortie peuvent être chiffrés et déchiffrés. Les sections suivantes décrivent comment AWS KMS est utilisé pour ces deux processus.

Rubriques

- Chiffrement du fichier d'entrée
- Déchiffrement du fichier d'entrée
- Chiffrement du fichier de sortie
- Protection du contenu HLS
- Contexte de chiffrement Elastic Transcoder

Chiffrement du fichier d'entrée

Avant de pouvoir utiliser Elastic Transcoder, vous devez créer un compartiment Amazon S3 et y télécharger votre fichier multimédia. Vous pouvez chiffrer le fichier avant de le télécharger à l'aide du chiffrement côté client AES ou après l'avoir téléchargé à l'aide du chiffrement côté serveur Amazon S3.

Si vous choisissez le chiffrement côté client d'AES, vous êtes responsable du chiffrement du fichier avant de le télécharger dans Amazon S3, et vous devez fournir à Elastic Transcoder l'accès à la clé de chiffrement. Pour cela, vous devez utiliser une AWS KMS key AWS KMS symétrique pour protéger la clé de chiffrement AES que vous avez utilisée pour chiffrer le fichier multimédia.

Si vous choisissez le chiffrement côté serveur, vous autorisez Amazon S3 à effectuer toutes les opérations de chiffrement et de déchiffrement des fichiers en votre nom. Vous pouvez configurer Amazon S3 pour utiliser l'un des trois types de clés de chiffrement pour protéger la clé de données unique utilisée pour chiffrer votre fichier :

- Une clé Amazon S3, une clé de chiffrement qu'Amazon S3 possède et gère. Elle ne fait pas partie de votre Compte AWS.
- La Clé gérée par AWS pour Amazon S3, une clé KMS qui fait partie de votre compte, mais qui est créée et gérée par AWS
- Toute clé gérée par le client symétrique que vous créez à l'aide de AWS KMS.

> **Important**
>
> Pour le chiffrement côté client et côté serveur, Elastic Transcoder ne prend en charge que les clés KMS symétriques. Vous ne pouvez pas utiliser de clé KMS asymétrique pour chiffrer vos fichiers Elastic Transcoder. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Vous pouvez activer le chiffrement et spécifier une clé à l'aide de la console Amazon S3 ou des API Amazon S3 appropriées. Pour plus d'informations sur la manière dont Amazon S3 applique le chiffrement, consultez Protection des données à l'aide du chiffrement côté serveur avec des clés KMS (SSE-KMS) dans le guide de l'utilisateur Amazon Simple Storage Service.

Lorsque vous protégez votre fichier d'entrée à l'aide de la Clé gérée par AWS pour Amazon S3 dans votre compte ou une clé gérée par le client, Amazon S3 et AWS KMS interagissent de la façon suivante :

1. Amazon S3 demande une clé de données en texte brut et une copie de la clé de données chiffrée sous la clé KMS spécifiée.
2. AWS KMS crée une clé de données, la chiffre avec la clé KMS spécifiée, puis envoie la clé de données en texte brut et la clé de données chiffrée à Amazon S3.
3. Amazon S3 utilise la clé de données en texte brut pour chiffrer le fichier multimédia, puis stocke le fichier dans le compartiment Amazon S3 spécifié.
4. Amazon S3 stocke la clé de données chiffrée avec le fichier multimédia chiffré.

Déchiffrement du fichier d'entrée

Si vous choisissez le chiffrement côté serveur d'Amazon S3 pour chiffrer le fichier d'entrée, Elastic Transcoder ne déchiffre pas le fichier. Au lieu de cela, Elastic Transcoder s'appuie sur Amazon S3

pour effectuer le déchiffrement en fonction des paramètres que vous spécifiez lorsque vous créez une tâche et un pipeline.

Les combinaisons suivantes de paramètres sont disponibles.

Mode de chiffrement	Clé AWS KMS	Signification
`S3`	Par défaut	Amazon S3 crée et gère les clés utilisées pour chiffrer et déchiffrer le fichier multimédia. Le processus est opaque pour l'utilisateur.
`S3-AWS-KMS`	Par défaut	Amazon S3 utilise une clé de données chiffrée par la Clé gérée par AWS par défaut pour Amazon S3 dans votre compte pour chiffrer le fichier multimédia.
`S3-AWS-KMS`	Personnalisée (avec nom ARN)	Amazon S3 utilise une clé de données chiffrée par la clé gérée par le client spécifiée pour chiffrer le fichier multimédia.

Lorsque `S3-AWS-KMS` est spécifié, Amazon S3 et AWS KMS collaborent comme indiqué ci-dessous pour effectuer le déchiffrement.

1. Amazon S3 envoie la clé de données chiffrée à AWS KMS.
2. AWS KMS déchiffre la clé de données à l'aide de la clé KMS appropriée, puis renvoie la clé de données en texte brut à Amazon S3.
3. Amazon S3 utilise la clé de données en texte brut pour déchiffrer le texte chiffré.

Si vous choisissez le chiffrement côté client à l'aide d'une clé AES, Elastic Transcoder récupère le fichier chiffré à partir du compartiment Amazon S3 et le déchiffre. Elastic Transcoder utilise la clé

KMS que vous avez spécifiée lorsque vous avez créé le pipeline pour déchiffrer la clé AES, puis utilise la clé AES pour déchiffrer le fichier multimédia.

Chiffrement du fichier de sortie

Elastic Transcoder chiffre le fichier de sortie en fonction de la façon dont vous spécifiez les paramètres de chiffrement lorsque vous créez une tâche et un pipeline. Les options suivantes sont disponibles.

Mode de chiffrement	Clé AWS KMS	Signification
`S3`	Par défaut	Amazon S3 crée et gère les clés utilisées pour chiffrer le fichier de sortie.
`S3-AWS-KMS`	Par défaut	Amazon S3 utilise une clé de données créée par AWS KMS et chiffrée par la Clé gérée par AWS pour Amazon S3 dans votre compte.
`S3-AWS-KMS`	Personnalisée (avec nom ARN)	Amazon S3 utilise une clé de données chiffrée à l'aide de la clé gérée par le client, spécifiée par l'ARN, pour chiffrer le fichier multimédia.
`AES-`	Par défaut	Elastic Transcoder utilise la Clé gérée par AWS pour Amazon S3 dans votre compte pour déchiffrer la clé AES spécifiée que vous fournissez, et utilise cette clé pour chiffrer le fichier de sortie.
`AES-`	Personnalisée (avec nom ARN)	Elastic Transcoder utilise la clé gérée par le client, spécifiée par l'ARN, pour

Mode de chiffrement	Clé AWS KMS	Signification
		déchiffrer la clé AES spécifiée que vous fournissez, et utilise cette clé pour chiffrer le fichier de sortie.

Lorsque vous spécifiez que la Clé gérée par AWS pour Amazon S3 dans votre compte ou qu'une clé gérée par le client doit être utilisée pour chiffrer le fichier de sortie, Amazon S3 et AWS KMS interagissent comme suit :

1. Amazon S3 demande une clé de données en texte brut et une copie de la clé de données chiffrée sous la clé KMS spécifiée.
2. AWS KMS crée une clé de données, la chiffre à l'aide de la clé KMS et envoie la clé de données en texte brut et la clé de données chiffrée à Amazon S3.
3. Amazon S3 chiffre le fichier multimédia à l'aide de la clé de données et le stocke dans le compartiment Amazon S3 spécifié.
4. Amazon S3 stocke la clé de données chiffrée avec le fichier multimédia chiffré.

Lorsque vous spécifiez que votre clé AES fournie doit être utilisée pour chiffrer le fichier de sortie, la clé AES doit être chiffrée à l'aide d'une clé KMS dans AWS KMS. Elastic Transcoder, AWS KMS, et vous, interagissez de la manière suivante :

1. Vous chiffrez votre clé AES en appelant l'opération Encrypt dans l'API AWS KMS. AWS KMS chiffre la clé en utilisant la clé KMS spécifiée. Vous spécifiez la clé KMS à utiliser lorsque vous créez le pipeline.
2. Vous spécifiez le fichier contenant la clé AES chiffrée lorsque vous créez la tâche Elastic Transcoder.
3. Elastic Transcoder déchiffre la clé en appelant l'opération Decrypt dans l'API AWS KMS, en transmettant la clé chiffrée en tant que texte chiffré.
4. Elastic Transcoder utilise la clé AES déchiffrée pour chiffrer le fichier multimédia de sortie, puis supprime la clé AES déchiffrée de la mémoire. Seule la copie chiffrée que vous avez définie initialement dans la tâche est enregistrée sur le disque.
5. Vous pouvez télécharger le fichier de sortie chiffré et le déchiffrer localement en utilisant la clé AES originale que vous avez définie.

> **Important**
>
> AWS ne stocke jamais vos clés de chiffrement privées. Par conséquent, il est important que vous gériez vos clés de façon fiable et sécurisée. Si vous les perdez, vous ne pourrez pas lire vos données.

Protection du contenu HLS

HTTP Live Streaming (HLS) est un protocole de streaming adaptatif. Elastic Transcoder prend en charge le protocole HLS en fractionnant votre fichier d'entrée en fichiers individuels plus petits, appelés segments multimédias. Un ensemble de segments multimédias individuels correspondants contient les mêmes éléments encodés selon différents taux d'échantillonnage, ce qui permet au lecteur de sélectionner le flux qui correspond le mieux à la bande passante disponible. Elastic Transcoder crée également des listes de lecture contenant des métadonnées pour les différents segments pouvant être diffusés.

Lorsque vous activez la protection de contenu HLS, chaque segment multimédia est chiffré à l'aide d'une clé de chiffrement AES-128. Lorsque le contenu est affiché, le lecteur télécharge la clé et déchiffre les segments multimédias au cours du processus de lecture.

Deux types de clés sont utilisés : une clé KMS et une clé de données. Vous devez créer une clé KMS pour chiffrer et déchiffrer la clé de données. Elastic Transcoder utilise la clé de données pour chiffrer et déchiffrer les segments multimédias. La clé de données doit être de type AES-128. Toutes les variations et tous les segments d'un même contenu sont chiffrés à l'aide de la même clé de données. Vous pouvez fournir une clé de données ou laisser Elastic Transcoder la créer pour vous.

La clé KMS peut être utilisée pour chiffrer la clé de données aux points suivants :

- Si vous fournissez votre propre clé de données, vous devez la chiffrer avant de la transmettre à Elastic Transcoder.
- Si vous demandez à Elastic Transcoder de générer la clé de données, Elastic Transcoder chiffre la clé de données pour vous.

La clé KMS peut être utilisée pour déchiffrer la clé de données aux points suivants :

- Elastic Transcoder déchiffre votre clé de données fournie lorsqu'il a besoin d'utiliser la clé de données pour chiffrer le fichier de sortie ou déchiffrer le fichier d'entrée.

- Vous déchiffrez une clé de données générée par Elastic Transcoder et l'utilisez pour déchiffrer les fichiers de sortie.

Pour de plus amples informations, veuillez consulter Protection du contenu HLS dans le Guide du développeur Amazon Elastic Transcoder.

Contexte de chiffrement Elastic Transcoder

Un contexte de chiffrement est un ensemble de paires clé-valeur qui contiennent des données non secrètes arbitraires. Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de données, AWS KMS lie de manière chiffrée le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez transmettre le même contexte de chiffrement.

Elastic Transcoder utilise le même contexte de chiffrement dans toutes les demandes d'API AWS KMS pour générer des clés de données, chiffrer et déchiffrer.

```
"service" : "elastictranscoder.amazonaws.com"
```

Le contexte de chiffrement est écrit dans CloudTrail les journaux pour vous aider à comprendre comment une clé AWS KMS KMS donnée a été utilisée. Dans le `requestParameters` champ d'un fichier CloudTrail journal, le contexte de chiffrement est similaire au suivant :

```
"encryptionContext": {
  "service" : "elastictranscoder.amazonaws.com"
}
```

Pour plus d'informations sur la façon de configurer les tâches Elastic Transcoder afin d'utiliser l'une des options de chiffrement prises en charge, veuillez consulter Options de chiffrement des données dans le Guide du développeur Amazon Elastic Transcoder.

Comment Amazon EMR utilise AWS KMS

Lorsque vous utilisez un cluster Amazon EMR, vous pouvez le configurer pour chiffrer les données au repos avant de les enregistrer dans un emplacement de stockage permanent. Vous pouvez chiffrer des données au repos dans le système de fichiers EMR (EMRFS), sur les volumes de stockage de nœuds de cluster, ou les deux. Pour chiffrer les données au repos, vous pouvez utiliser une AWS KMS key. Les rubriques suivantes expliquent comment un cluster Amazon EMR utilise une clé KMS pour chiffrer des données au repos.

> **Important**
>
> Amazon EMR prend uniquement en charge les clés KMS symétriques. Vous ne pouvez pas utiliser une clé KMS asymétrique pour chiffrer les données au repos dans un cluster Amazon EMR. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Les clusters Amazon EMR chiffrent également les données en transit, ce qui signifie que le cluster chiffre les données avant de les envoyer via le réseau. Vous ne pouvez pas utiliser une clé KMS pour chiffrer des données en transit. Pour de plus amples informations, veuillez consulter Chiffrement des données en transit dans le guide de gestion Amazon EMR.

Pour plus d'informations sur toutes les options de chiffrement disponibles dans Amazon EMR, veuillez consulter Options de chiffrement dans le guide de gestion Amazon EMR.

Rubriques

- Chiffrement des données dans le système de fichiers EMR (EMRFS)
- Chiffrement des données sur les volumes de stockage de nœuds de Cluster
- Contexte de chiffrement

Chiffrement des données dans le système de fichiers EMR (EMRFS)

Les clusters Amazon EMR utilisent deux systèmes de fichiers distribués :

- Le système de fichiers distribué Hadoop (HDFS). Le chiffrement HDFS n'utilise pas une clé KMS dans AWS KMS.
- Le système de fichiers EMR (EMRFS). EMRFS est une implémentation de HDFS, qui permet aux clusters Amazon EMR de stocker des données dans Amazon Simple Storage Service (Amazon S3). EMRFS prend en charge quatre options de chiffrement, dont deux utilisent une clé KMS dans AWS KMS. Pour plus d'informations sur les quatre options de chiffrement EMRFS, veuillez consulter Options de chiffrement dans le guide de gestion Amazon EMR.

Les deux options de chiffrement EMRFS qui utilisent une clé KMS font appel aux fonctions de chiffrement suivantes proposées par Amazon S3 :

- Protection des données grâce au chiffrement côté serveur avec AWS Key Management Service (SSE-KMS). Le cluster Amazon EMR envoie les données à Simple Storage Service (Amazon S3). Simple Storage Service (Amazon S3) utilise une clé KMS pour chiffrer les données avant de les enregistrer dans un compartiment S3. Pour en savoir plus sur la façon dont cela fonctionne, veuillez consulter Processus de chiffrement des données sur EMRFS avec SSE-KMS.
- Protection des données via le chiffrement côté client (CSE-KMS). Les données d'un Amazon EMR sont chiffrées sous une AWS KMS key avant d'être envoyées à Simple Storage Service (Amazon S3) pour y être stockées. Pour en savoir plus sur la façon dont cela fonctionne, veuillez consulter Processus de chiffrement des données sur EMRFS avec CSE-KMS.

Lorsque vous configurez un cluster Amazon EMR pour chiffrer les données sur EMRFS avec une clé KMS, vous choisissez la clé KMS que vous voulez que le cluster Amazon EMR ou Simple Storage Service (Amazon S3) utilise. Avec SSE-KMS, vous pouvez choisir la Clé gérée par AWS pour Amazon S3 avec l'alias aws/s3, ou une clé symétrique gérée par le client que vous créez. Avec le chiffrement côté client, vous devez choisir une clé symétrique gérée par le client que vous créez. Lorsque vous choisissez une clé gérée par le client, vous devez vous assurer que votre cluster Amazon EMR est autorisé à utiliser la clé KMS. Pour de plus amples informations, veuillez consulter Utilisation des AWS KMS keys pour le chiffrement dans le guide de gestion Amazon EMR.

Pour le chiffrement côté serveur et côté client, la clé KMS que vous choisissez est la clé racine dans un flux de chiffrement d'enveloppe. Les données sont chiffrées avec une clé de données unique qui est chiffrée sous la clé KMS dans AWS KMS. Les données chiffrées et une copie chiffrée de leur clé de données sont stockées ensemble en tant qu'objet chiffré unique dans un compartiment S3. Pour plus d'informations sur la façon dont cela fonctionne, consultez les rubriques suivantes.

Rubriques

- Processus de chiffrement des données sur EMRFS avec SSE-KMS
- Processus de chiffrement des données sur EMRFS avec CSE-KMS

Processus de chiffrement des données sur EMRFS avec SSE-KMS

Lorsque vous configurez un cluster Amazon EMR pour utiliser SSE-KMS, le processus de chiffrement fonctionne comme suit :

1. Le cluster envoie les données à Amazon S3 pour leur stockage dans un compartiment S3.

2. Amazon S3 envoie une GenerateDataKeydemande àAWS KMS, en spécifiant l'ID de clé KMS que vous avez choisi lorsque vous avez configuré le cluster pour utiliser SSE-KMS. La demande inclut le contexte de chiffrement. Pour plus d'informations, veuillez consulter Contexte de chiffrement.
3. AWS KMS génère une clé de chiffrement des données unique (clé de données), puis envoie deux copies de cette clé de données à Amazon S3. Une copie est non chiffrée (texte brut) et l'autre copie est chiffrée sous la clé KMS.
4. Amazon S3 utilise la clé de données en texte brut pour chiffrer les données reçues à l'étape 1, puis supprime la clé de données en texte brut de la mémoire dès que possible après usage.
5. Amazon S3 stocke ensemble les données chiffrées et la copie chiffrée de la clé de données en tant qu'objet chiffré unique dans un compartiment S3.

Le processus de déchiffrement fonctionne comme suit :

1. Le cluster demande un objet de données chiffré depuis un compartiment S3.
2. Amazon S3 extrait la clé de données chiffrée de l'objet S3, puis l'envoie à AWS KMS avec une demande Decrypt. La demande comprend un contexte de chiffrement.
3. AWS KMS déchiffre la clé de données chiffrée à l'aide de la clé KMS utilisée pour la chiffrer, puis envoie la clé de données déchiffrée (texte brut) à Amazon S3.
4. Amazon S3 utilise la clé de données en texte brut pour déchiffrer les données chiffrées, puis supprime la clé de données en texte brut de la mémoire dès que possible après usage.
5. Amazon S3 envoie les données déchiffrées au cluster.

Processus de chiffrement des données sur EMRFS avec CSE-KMS

Lorsque vous configurez un cluster Amazon EMR pour utiliser CSE-KMS, le processus de chiffrement fonctionne comme suit :

1. Lorsqu'il est prêt à stocker des données dans Amazon S3, le cluster envoie une GenerateDataKeydemande àAWS KMS, en spécifiant l'ID de clé KMS que vous avez choisi lorsque vous avez configuré le cluster pour utiliser CSE-KMS. La demande inclut le contexte de chiffrement. Pour plus d'informations, veuillez consulter Contexte de chiffrement.
2. AWS KMS génère une clé de chiffrement de données unique (clé de données), puis envoie deux copies de cette clé de données au cluster. Une copie est non chiffrée (texte brut) et l'autre copie est chiffrée sous la clé KMS.

3. Le cluster utilise la clé de données en texte brut pour chiffrer les données, puis supprime la clé de données en texte brut de la mémoire dès que possible après usage.
4. Le cluster combine les données chiffrées et la copie chiffrée de la clé de données en un objet chiffré unique.
5. Le cluster envoie l'objet chiffré à Amazon S3 pour stockage.

Le processus de déchiffrement fonctionne comme suit :

1. Le cluster demande l'objet de données chiffré depuis un compartiment S3.
2. Amazon S3 envoie l'objet chiffré au cluster.
3. Le cluster extrait la clé de données chiffrée de l'objet chiffré, puis l'envoie à AWS KMS avec une demande Decrypt. La demande inclut le contexte de chiffrement.
4. AWS KMS déchiffre la clé de données chiffrée à l'aide de la clé KMS utilisée pour la chiffrer, puis envoie la clé de données déchiffrée (texte brut) au cluster.
5. Le cluster utilise la clé de données en texte brut pour déchiffrer les données chiffrées, puis supprime la clé de données en texte brut de la mémoire dès que possible après usage.

Chiffrement des données sur les volumes de stockage de nœuds de Cluster

Un cluster Amazon EMR est une collection d'instances Amazon Elastic Compute Cloud (Amazon EC2). Chaque instance du cluster est appelée nœud de cluster ou nœud. Chaque nœud peut avoir deux types de volumes de stockage : des volumes de stockage d'instance et des volumes Amazon Elastic Block Store (Amazon EBS). Vous pouvez configurer le cluster pour utiliser Linux Unified Key Setup (LUKS) pour chiffrer les deux types de volumes de stockage sur les nœuds (mais pas le volume de démarrage de chaque nœud). Il s'agit du chiffrement de disque local.

Lorsque vous activez le chiffrement de disque local pour un cluster, vous pouvez choisir de chiffrer la clé LUKS avec une clé KMS dans AWS KMS. Vous devez choisir une clé gérée par le client que vous créez ; vous ne pouvez pas utiliser une Clé gérée par AWS. Si vous choisissez une clé gérée par le client, vous devez vous assurer que votre cluster Amazon EMR est autorisé à utiliser la clé KMS. Pour de plus amples informations, veuillez consulter Utilisation des AWS KMS keys pour le chiffrement dans le guide de gestion Amazon EMR.

Lorsque vous activez le chiffrement de disque local à l'aide d'une clé KMS, le processus de chiffrement fonctionne comme ceci :

1. Lorsque chaque nœud de cluster est lancé, il envoie une GenerateDataKeydemande àAWS KMS, spécifiant l'ID de clé KMS que vous avez choisi lorsque vous avez activé le chiffrement de disque local pour le cluster.
2. AWS KMS génère une clé de chiffrement de données unique (clé de données), puis envoie deux copies de cette clé de données au nœud. Une copie est non chiffrée (texte brut) et l'autre copie est chiffrée sous la clé KMS.
3. Le nœud utilise une version encodée en base 64 de la clé de données en texte brut comme mot de passe qui protège la clé LUKS. Le nœud enregistre la copie chiffrée de la clé de données sur le volume de démarrage.
4. Si le nœud redémarre, le nœud redémarré envoie la clé de données chiffrée à AWS KMS avec une demande Decrypt.
5. AWS KMS déchiffre la clé de données chiffrée à l'aide de la clé KMS utilisée pour la chiffrer, puis envoie la clé de données déchiffrée (texte brut) au nœud.
6. Le nœud utilise la version encodée en base 64 de la clé de données en texte brut comme mot de passe pour déverrouiller la clé LUKS.

Contexte de chiffrement

Chaque service AWS intégré à AWS KMS peut spécifier un contexte de chiffrement quand il utilise AWS KMS pour générer des clés de données, ou pour chiffrer ou déchiffrer des données. Le contexte de chiffrement représente des informations authentifiées supplémentaires qu'AWS KMS utilise pour vérifier l'intégrité des données. Quand service spécifie un contexte de chiffrement pour une opération de chiffrement, il doit spécifier le même contexte de chiffrement pour l'opération de déchiffrement correspondante, sinon le déchiffrement échouera. Le contexte de chiffrement est également écrit dans les fichiers journaux AWS CloudTrail qui vous aident à comprendre pourquoi une clé KMS spécifique a été utilisée.

La section suivante décrit le contexte de chiffrement utilisé dans chaque scénario de chiffrement Amazon EMR qui utilise une clé KMS.

Contexte de chiffrement pour le chiffrement EMRFS avec SSE-KMS

Avec SSE-KMS, le cluster Amazon EMR envoie les données à Amazon S3, puis Amazon S3 utilise une clé KMS pour chiffrer les données avant de les enregistrer dans un compartiment S3. Dans ce cas, Amazon S3 utilise le nom de ressource Amazon (ARN) de l'objet S3 comme contexte de chiffrement pour chaque demande GenerateDataKeyde déchiffrement à AWS KMS laquelle il envoie. L'exemple suivant montre une représentation JSON du contexte de chiffrement qu'Amazon S3 utilise.

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Contexte de chiffrement pour le chiffrement EMRFS avec CSE-KMS

Avec CSE-KMS, le cluster Amazon EMR utilise une clé KMS pour chiffrer les données avant de les envoyer à Amazon S3 pour stockage. Dans ce cas, le cluster utilise l'Amazon Resource Name (ARN) de la clé KMS comme contexte de chiffrement pour chaque GenerateDataKeydemande de déchiffrement à AWS KMS laquelle il envoie. L'exemple suivant montre une représentation JSON du contexte de chiffrement que le cluster utilise.

```
{ "kms_cmk_id" : "arn:aws:kms:us-
east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Contexte de chiffrement pour le chiffrement de disque Local avec LUKS

Lorsqu'un cluster Amazon EMR utilise le chiffrement de disque local avec LUKS, les nœuds du cluster ne spécifient pas le contexte de chiffrement des demandes GenerateDataKeyet de déchiffrement auxquelles ils envoient. AWS KMS

Comment AWS Nitro Enclaves utilise AWS KMS

AWS KMS prend en charge l'attestation cryptographique pour les enclaves Nitro AWS. Les applications qui prennent en charge les enclaves Nitro AWS exécutent les opérations AWS KMS cryptographiques suivantes avec un document d'attestation signé pour l'enclave. Ces API AWS KMS vérifient que le document d'attestation provient d'une enclave Nitro. Ensuite, au lieu de renvoyer des données en texte brut dans la réponse, ces API chiffrent le texte brut avec la clé publique du document d'attestation et renvoient un texte chiffré uniquement par la clé privée correspondante dans l'enclave.

- Decrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

Le tableau suivant montre en quoi la réponse aux demandes d'enclave Nitro diffère de la réponse standard pour chaque opération d'API.

Opération AWS KMS	Réponse normale	Réponse pour les enclaves Nitro AWS
`Decrypt`	Renvoie des données en texte brut	Renvoie les données en texte brut par la clé publique à partir du document d'attestation
`GenerateDataKey`	Renvoie une copie en texte brut de la clé de données (Renvoie également une copie de la clé de données avec une clé KMS)	Renvoie une copie de la clé de données chiffrées par la clé publique à partir du document d'attestation (Renvoie également une copie de la clé de données avec une clé KMS)
`GenerateDataKeyPair`	Renvoie une copie en texte brut de la clé privée (Renvoie également la clé publique et une copie de la clé privée chiffrée avec une clé KMS)	Renvoie une copie de la clé privée chiffrée par la clé publique à partir du document d'attestation (Renvoie également la clé publique et une copie de la clé privée chiffrée avec une clé KMS)
`GenerateRandom`	Renvoie une chaîne d'octets aléatoire	Renvoie les chaîne d'octets aléatoire chiffrée par la clé publique à partir du document d'attestation

AWS KMS prend en charge les clés de condition de stratégie que vous pouvez utiliser pour autoriser ou refuser les opérations d'enclave sur une clé AWS KMS uniquement lorsque le document d'attestation contient le contenu spécifié. Vous pouvez également suivre les demandes AWS KMS relatives à votre enclave Nitro dans vos journaux AWS CloudTrail.

Rubriques

- Comment appeler des API AWS KMS pour une enclave Nitro
- Clés de condition AWS KMS pour AWS Nitro Enclaves
- Demandes de surveillance pour les enclaves Nitro

Comment appeler des API AWS KMS pour une enclave Nitro

Pour appeler des API AWS KMS pour une enclave Nitro, utilisez le paramètre `Recipient` de la demande pour fournir le document d'attestation signé pour l'enclave et l'algorithme de chiffrement à utiliser avec la clé publique de l'enclave. Lorsqu'une demande inclut le paramètre `Recipient` avec un document d'attestation signé, la réponse inclut un champ `CiphertextForRecipient` contenant le texte chiffré par la clé publique. Le champ de texte brut est nul ou vide.

Le paramètre `Recipient` doit spécifier un document d'attestation signé provenant d'une enclave AWS Nitro. AWS KMS s'appuie sur la signature numérique du document d'attestation de l'enclave pour prouver que la clé publique de la demande provenait d'une enclave valide. Vous ne pouvez pas fournir votre propre certificat pour signer numériquement le document d'attestation.

Pour spécifier le paramètre `Recipient`, utilisez le SDK d’enclaves Nitro AWS ou n’importe quel SDK AWS. Le SDK d’enclaves Nitro AWS, qui n'est pris en charge que dans une enclave Nitro, ajoute automatiquement le paramètre `Recipient` et ses valeurs à chaque demande AWS KMS. Pour demander des enclaves Nitro dans les SDK AWS, vous devez spécifier le paramètre `Recipient` et ses valeurs. Support de l'attestation cryptographique de l'enclave Nitro dans les SDK AWS a été introduit en mars 2023.

AWS KMS prend en charge les clés de condition de stratégie que vous pouvez utiliser pour autoriser ou refuser les opérations d'enclave sur une clé AWS KMS uniquement lorsque le document d'attestation contient le contenu spécifié. Vous pouvez également suivre les demandes AWS KMS relatives à votre enclave Nitro dans vos journaux AWS CloudTrail.

Pour obtenir des informations détaillées sur le `Recipient` paramètre et le champ de `CiphertextForRecipient` réponse AWS, consultez les GenerateRandomrubriques Déchiffrer, GenerateDataKeyGenerateDataKeyPair, et dans la référence des AWS Key Management Service API, le SDK AWSNitro Enclaves ou tout autre SDK. AWS Pour plus d'informations sur la configuration de vos données et clés de données pour le chiffrement, veuillez consulter Using cryptographic attestation with AWS KMS.

Clés de condition AWS KMS pour AWS Nitro Enclaves

Vous pouvez spécifier les clés de conditions dans les stratégies de clé et les politiques IAM qui contrôlent l'accès à vos ressources AWS KMS. Les déclarations de politique qui incluent une clé de condition ne sont efficaces que lorsque ses conditions sont satisfaites.

AWS KMSfournit des clés de condition qui limitent les autorisations pour les GenerateRandomopérations de déchiffrement GenerateDataKeyGenerateDataKeyPair,, et en fonction du contenu du document d'attestation signé dans la demande. Ces clés de condition ne fonctionnent que lorsqu'une demande d'opération AWS KMS inclut le paramètre `Recipient` avec un document d'attestation valide provenant d'une enclave Nitro AWS. Pour spécifier le paramètre `Recipient`, utilisez le SDK d'enclaves Nitro AWS ou n'importe quel SDK AWS.

Cette clé de condition AWS KMS spécifique à l'enclave est valide dans les instructions de politique de clé et les instructions de politique IAM, même si elle n'apparaît pas dans la console IAM ou dans la référence d'autorisations de service IAM.

km RecipientAttestation : ImageSha 384

Clés de condition AWS KMS	Type de condition	Type de la valeur	Opérations d'API	Type de politique

`kms:RecipientAttestation:ImageSha384`	Chaîne	À valeur unique	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Politiques de clé et politiques IAM

La clé de condition `kms:RecipientAttestation:ImageSha384` contrôle l'accès à `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair` et `GenerateRandom` avec une clé KMS lorsque le résumé d'image du document d'attestation signé dans la demande correspond à la valeur de la clé de condition. La valeur `ImageSha384` correspond au PCR0 du document d'attestation. Cette clé de

condition n'est effective que lorsque le paramètre `Recipient` de la demande spécifie un document d'attestation signé pour une enclave AWS Nitro.

Cette valeur est également incluse dans les CloudTrailévénements relatifs aux demandes adressées AWS KMS aux enclaves Nitro.

> **Note**
>
> Cette clé de condition est valide dans les instructions de politique de clé et les instructions de politique IAM, même si elle n'apparaît pas dans la console IAM ou dans la référence d'autorisations de service IAM.

Par exemple, la déclaration de politique clé suivante autorise le `data-processing` rôle à utiliser la clé KMS pour les GenerateRandomopérations de déchiffrement GenerateDataKey, GenerateDataKeyPair, et. La clé de condition `kms:RecipientAttestation:ImageSha384` permet les opérations uniquement lorsque la valeur de hachage de l'image (PCR0) du document d'attestation de la demande correspond à la valeur de hachage de l'image de la condition. Cette clé de condition n'est effective que lorsque le paramètre `Recipient` de la demande spécifie un document d'attestation signé pour une enclave AWS Nitro.

Si la demande n'inclut pas de document d'attestation valide provenant d'une enclave AWS Nitro, l'autorisation est refusée car cette condition n'est pas remplie.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
```

```
    }
  }
}
```

km : : PCR RecipientAttestation <PCR_ID>

Clés de condition AWS KMS	Type de condition	Type de la valeur	Opérations d'API	Type de politique

`kms:RecipientAttestation:PCR<PCR_ID>`	Chaîne	À valeur unique	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Politiques de clé et politiques IAM

La clé de condition `kms:RecipientAttestation:PCR<PCR_ID>` contrôle l'accès à `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair` et `GenerateRandom` avec une clé KMS uniquement lorsque les registres de configuration de plateforme (PCR) du document d'attestation signé de la demande correspondent aux PCR de la clé de condition. Cette clé de condition n'est effective que lorsque le paramètre `Recipient` de la demande spécifie un document d'attestation signé pour une enclave AWS Nitro.

Cette valeur est également incluse dans les CloudTrailévénements qui représentent des demandes adressées à AWS KMS des enclaves Nitro.

> **Note**
>
> Cette clé de condition est valide dans les instructions de politique de clé et les instructions de politique IAM, même si elle n'apparaît pas dans la console IAM ou dans la référence d'autorisations de service IAM.

Pour spécifier une valeur PCR, utilisez le format suivant. Concaténez l'ID de PCR au nom de clé de condition. La valeur PCR doit être une chaîne hexadécimale en minuscules de 96 octets maximum.

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```

Par exemple, la clé de condition suivante spécifie une valeur particulière pour PCR1, qui correspond au hachage du noyau utilisé pour l'enclave et le processus d'amorçage.

```
kms:RecipientAttestation:PCR1:
 "0x1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcdef
```

Par exemple, l'instruction de stratégie de clé suivante autorise le rôle `data-processing` à utiliser la clé KMS pour l'opération Decrypt.

La clé de condition `kms:RecipientAttestation:PCR` de cette instruction autorise l'opération uniquement lorsque la valeur PCR1 du document d'attestation signé dans la demande correspond à la valeur `kms:RecipientAttestation:PCR1` de la condition. Utilisez l'opérateur de politique `StringEqualsIgnoreCase` pour exiger une comparaison insensible à la casse des valeurs PCR.

Si la demande n'inclut pas de document d'attestation, l'autorisation est refusée car cette condition n'est pas remplie.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
 "0x1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f15
    }
  }
}
```

Demandes de surveillance pour les enclaves Nitro

Vous pouvez utiliser vos AWS CloudTrail journaux pour surveiller le déchiffrement, GenerateDataKeyGenerateDataKeyPair, et les GenerateRandomopérations d'une enclave AWS Nitro. Dans ces entrées de journal, le champ `additionalEventData` contient un champ `recipient` contenant l'ID du module (`attestationDocumentModuleId`), le condensé d'image (`attestationDocumentEnclaveImageDigest`) et les registres de configuration de plateforme (PCR) provenant du document d'attestation contenu dans la demande. Ces champs sont uniquement inclus lorsque le paramètre `Recipient` de la demande spécifie un document d'attestation signé pour une enclave Nitro AWS.

L'ID du module est l'ID d'enclave de l'enclave Nitro. Le résumé d'image est le hachage SHA384 de l'image d'enclave. Vous pouvez utiliser le résumé de l'image et les valeurs PCR dans les conditions des stratégies de clé et des politiques IAM. Pour plus d'informations sur les PCR, consultez Où obtenir les mesures d'une enclave dans le Guide de l'utilisateur des enclaves Nitro AWS.

Cette section présente un exemple d'entrée de CloudTrail journal pour chacune des demandes d'enclave Nitro prises en charge àAWS KMS.

Decrypt (pour une enclave)

L'exemple suivant illustre une entrée de journal AWS CloudTrail pour l'opération Decrypt d'une enclave Nitro AWS.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
```

```
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
    "eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey (pour une enclave)

L'exemple suivant montre une entrée dans le AWS CloudTrail journal d'une GenerateDataKeyopération pour une enclave AWS Nitro.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
```

```
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair (pour une enclave)

L'exemple suivant montre une entrée dans le AWS CloudTrail journal d'une GenerateDataKeyPairopération pour une enclave AWS Nitro.

```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateRandom (pour une enclave)

L'exemple suivant montre une entrée dans le AWS CloudTrail journal d'une GenerateRandomopération pour une enclave AWS Nitro.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Comment d'Amazon Redshift utilise AWS KMS

Cette rubrique explique comment Amazon Redshift utilise AWS KMS pour chiffrer des données.

Rubriques

- Chiffrement Amazon Redshift
- Contexte de chiffrement

Chiffrement Amazon Redshift

Un entrepôt de données Amazon Redshift est un ensemble de ressources informatiques appelées nœuds, qui sont organisées en un groupe appelé cluster. Chaque cluster exécute un moteur Amazon Redshift et contient une ou plusieurs bases de données.

Amazon Redshift utilise une architecture à quatre niveaux de clés pour le chiffrement. Cette architecture se compose de clés de chiffrement des données, d'une clé de base de données, d'une clé de cluster et d'une clé racine. Vous pouvez utiliser une AWS KMS key comme clé racine.

Les clés de chiffrement de données chiffrent les blocs de données contenus dans le cluster. Chaque bloc de données se voit attribuer une clé AES-256 générée de façon aléatoire. Ces clés sont chiffrées à l'aide de la clé de base de données du cluster.

La clé de base de données chiffre les clés de chiffrement de données contenues dans le cluster. La clé de base de données est une clé AES-256 générée de façon aléatoire. Elle est stockée sur disque dans un autre réseau que celui du cluster Amazon Redshift et transmise au cluster via un canal sécurisé.

La clé de cluster chiffre la clé de base de données du cluster Amazon Redshift. Vous pouvez utiliser AWS KMS, AWS CloudHSM ou un module de sécurité matérielle (module HSM) externe pour gérer la clé de cluster. Consultez la documentation relative au chiffrement de base de données Amazon Redshift pour plus d'informations.

Vous pouvez demander le chiffrement en cochant la case appropriée dans la console Amazon Redshift. Vous pouvez spécifier une clé gérée par le client en la choisissant dans la liste qui s'affiche

sous la zone de chiffrement. Si vous ne spécifiez pas de clé gérée par le client, Amazon Redshift utilise la Clé gérée par AWS pour Amazon Redshift sous votre compte.

> **Important**
>
> Amazon Redshift prend uniquement en charge les clés KMS de chiffrement symétriques. Vous ne pouvez pas utiliser une clé KMS asymétrique dans un flux de travail de chiffrement Amazon Redshift. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, consultez Identification des clés KMS asymétriques.

Contexte de chiffrement

Chaque service intégré à AWS KMS spécifie un contexte de chiffrement lors de la demande de clés de données, du chiffrement et du déchiffrement de données. Le contexte de chiffrement représente des informations authentifiées supplémentaires qu'AWS KMS utilise pour vérifier l'intégrité des données. Autrement dit, lorsqu'un contexte de chiffrement est spécifié pour une opération de chiffrement, le service le spécifie également pour l'opération de déchiffrement, ou le déchiffrement échoue. Amazon Redshift utilise l'ID de cluster et le temps de création du contexte de chiffrement. Dans le `requestParameters` champ d'un fichier CloudTrail journal, le contexte de chiffrement sera similaire à celui-ci.

```
"encryptionContext": {
    "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",
    "aws:redshift:createtime": "20150206T1832Z"
},
```

Vous pouvez effectuer une recherche sur le nom du cluster dans vos CloudTrail journaux pour comprendre quelles opérations ont été effectuées à l'aide d'une AWS KMS key (clé KMS). Les opérations incluent le chiffrement du cluster, le déchiffrement du cluster et la génération de clés de données.

Comment Amazon Relational Database Service (Amazon RDS) utilise AWS KMS

Vous pouvez utiliser Amazon Relational Database Service (Amazon RDS) pour configurer, exploiter et mettre à l'échelle une base de données relationnelle dans le cloud. Vous pouvez chiffrer vos

ressources Amazon RDS à l'aide d'une Clé gérée par AWS ou d'une clé gérée par le client. Amazon RDS repose sur le chiffrement Amazon Elastic Block Store (Amazon EBS) pour assurer le chiffrement intégral des volumes de base de données.

Pour obtenir des informations détaillées sur la façon dont Amazon RDS utilise les clés KMS pour protéger vos ressources, consultez Encrypting Amazon RDS resources (Chiffrement des ressources Amazon RDS) et AWS KMS key management (Gestion des clés) dans le Guide de l'utilisateur Amazon RDS.

Comment AWS Secrets Manager utilise AWS KMS

AWS Secrets Manager est un service AWS qui chiffre et stocke vos secrets, les déchiffre de manière transparente et vous les renvoie en texte brut. Il est conçu spécialement pour stocker des secrets d'applications, tels que les informations d'identification de connexion, qui changent régulièrement et ne doivent pas être codées de manière irréversible, ni stockés sous forme de texte brut dans l'application. En lieu et place de recherches d'informations d'identification codées en dur ou de table, votre application appelle Secrets Manager.

Secrets Manager prend également en charge des fonctions qui font tourner régulièrement les secrets associés aux bases de données couramment utilisées. Il chiffre toujours les secrets ayant été récemment tournés avant qu'ils soient stockés.

Secrets Manager s'intègre à AWS Key Management Service (AWS KMS) pour chiffrer chaque version de chaque valeur de secret avec une clé de données unique protégée par une AWS KMS key. Cette intégration protège vos secrets avec des clés de chiffrement qui ne quittent jamais AWS KMS sous une forme non chiffrée. Elle vous permet également de définir des autorisations personnalisées sur la clé KMS et d'auditer les opérations qui génèrent, chiffrent et déchiffrent les clés de données utilisées pour protéger vos secrets.

Pour plus d'informations sur la façon dont Secrets Manager utilise les clés KMS pour protéger vos secrets, consultez Chiffrement et déchiffrement des secrets dans le Guide de l'utilisateur AWS Secrets Manager.

Comment Amazon Simple Email Service (Amazon SES) utilise AWS KMS

Vous pouvez utiliser Amazon Simple Email Service (Amazon SES) pour recevoir des e-mails et (éventuellement) pour chiffrer les messages électroniques reçus avant de les stocker dans un

compartiment Amazon Simple Storage Service (Amazon S3) de votre choix. Lorsque vous configurez Amazon SES pour chiffrer des messages électroniques, vous devez choisir la AWS KMS key AWS KMS sous laquelle Amazon SES chiffrera les messages. Vous pouvez choisir la Clé gérée par AWS pour Amazon SES (son alias est `aws/ses`), ou vous pouvez choisir une clé gérée par le client que vous avez créée dans AWS KMS.

> **Important**
>
> Amazon SES prend uniquement en charge les clés KMS symétriques. Vous ne pouvez pas utiliser une clé KMS asymétrique pour chiffrer vos messages électroniques Amazon SES. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Pour en savoir plus sur la réception d'e-mails à l'aide d'Amazon SES, veuillez consulter Réception d'e-mails avec Amazon SES dans le Guide du développeur Amazon Simple Email Service.

Rubriques

- Présentation du chiffrement Amazon SES à l'aide d'AWS KMS
- Contexte du chiffrement Amazon SES
- Autoriser Amazon SES à utiliser votre AWS KMS key
- Récupération et déchiffrement des messages électroniques

Présentation du chiffrement Amazon SES à l'aide d'AWS KMS

Lorsque vous configurez Amazon SES pour recevoir des e-mails et chiffrer les e-mails avant de les enregistrer dans votre compartiment S3, le processus fonctionne comme suit :

1. Vous créez une règle de réception pour Amazon SES, en spécifiant l'action S3, un compartiment S3 pour le stockage et une AWS KMS key pour le chiffrement.
2. Amazon SES reçoit un e-mail qui correspond à votre règle de réception.
3. Amazon SES demande une clé de données unique chiffrée avec la clé KMS que vous avez spécifiée dans la règle de réception applicable.
4. AWS KMS crée une nouvelle clé de données, la chiffre avec la clé KMS spécifiée, puis envoie les copies chiffrées et en texte brut de la clé de données à Amazon SES.

5. Amazon SES utilise la clé de données en texte brut pour chiffrer le message électronique, puis supprime la clé de données en texte brut de la mémoire dès que possible après usage.
6. Amazon SES place le message électronique chiffré et la clé de données chiffrée dans le compartiment S3 spécifié. La clé de données chiffrée est stockée sous forme de métadonnées avec le message électronique chiffré.

Pour accomplir Step 3 via Step 6, Amazon SES utilise le client de chiffrement Amazon S3 fourni par AWS. Utilisez le même client pour récupérer vos messages électroniques chiffrés à partir d'Amazon S3 et les déchiffrer. Pour plus d'informations, consultez Récupération et déchiffrement des messages électroniques.

Contexte du chiffrement Amazon SES

Quand Amazon SES demande une clé de données pour chiffrer les e-mails que vous avez reçus (Step 3 dans Présentation du chiffrement Amazon SES à l'aide d'AWS KMS), il inclut le contexte de chiffrement dans la demande. Le contexte de chiffrement fournit des données authentifiées supplémentaires (AAD) qu'AWS KMS utilise pour garantir l'intégrité des données. Le contexte de chiffrement est également écrit dans les fichiers journaux AWS CloudTrail qui vous aident à comprendre pourquoi une AWS KMS key (clé KMS) donnée a été utilisée. Amazon SES utilise le contexte de chiffrement suivant :

- L'ID du Compte AWS dans lequel vous avez configuré Amazon SES pour recevoir les messages électroniques
- Le nom de la règle de réception Amazon SES qui a appelé l'action S3 sur le message électronique
- L'ID de message Amazon SES pour le message électronique

L'exemple suivant montre une représentation JSON du contexte de chiffrement qu'Amazon SES utilise :

```
{
  "aws:ses:source-account": "111122223333",
  "aws:ses:rule-name": "example-receipt-rule-name",
  "aws:ses:message-id": "d6iitobk75ur44p8kdnnp7g2n800"
}
```

Autoriser Amazon SES à utiliser votre AWS KMS key

Pour chiffrer vos messages électroniques, vous pouvez utiliser la Clé gérée par AWS dans votre compte pour Amazon SES (`aws/ses`) ou une clé gérée par le client que vous créez. Amazon SES a déjà l'autorisation d'utiliser la Clé gérée par AWS en votre nom. Toutefois, si vous spécifiez une clé gérée par le client lorsque vous ajoutez l'action S3 à votre règle de réception Amazon SES, vous devez autoriser Amazon SES à utiliser la clé KMS pour chiffrer vos messages électroniques.

Pour accorder à Amazon SES l'autorisation d'utiliser votre clé gérée par le client, ajoutez l'instruction suivante à la politique de clé de votre clé KMS :

```
{
  "Sid": "Allow SES to encrypt messages using this KMS key",
  "Effect": "Allow",
  "Principal": {"Service": "ses.amazonaws.com"},
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:ses:rule-name": false,
      "kms:EncryptionContext:aws:ses:message-id": false
    },
    "StringEquals": {"kms:EncryptionContext:aws:ses:source-account": "ACCOUNT-ID-
WITHOUT-HYPHENS"}
  }
}
```

Remplacez *`ACCOUNT-ID-WITHOUT-HYPHENS`* par l'ID à 12 chiffres du Compte AWS dans lequel vous avez configuré Amazon SES pour recevoir les messages électroniques. Cette instruction de politique autorise Amazon SES à chiffrer les données avec cette clé KMS uniquement dans les conditions suivantes :

- Amazon SES doit spécifier `aws:ses:rule-name` et `aws:ses:message-id` dans le contexte `EncryptionContext` de leurs demandes d'API AWS KMS.
- Amazon SES doit spécifier `aws:ses:source-account` dans le contexte `EncryptionContext` de leurs demandes d'API AWS KMS, et la valeur pour `aws:ses:source-account` doit correspondre à l'ID de Compte AWS spécifié dans la politique de clé.

Pour plus d'informations sur le contexte de chiffrement utilisé par Amazon SES lors du chiffrement de vos e-mails, veuillez consulter Contexte du chiffrement Amazon SES. Pour obtenir des informations générales sur la façon dont AWS KMS utilise le contexte de chiffrement, consultez le contexte de chiffrement.

Récupération et déchiffrement des messages électroniques

Amazon SES n'a pas l'autorisation de déchiffrer vos messages électroniques chiffrés et ne peut pas les déchiffrer pour vous. Vous devez écrire du code pour récupérer vos e-mails depuis Amazon S3 et les déchiffrer. Pour simplifier cette opération, utilisez le client de chiffrement Amazon S3. Les kits SDK AWS suivants incluent le client de chiffrement Amazon S3 :

- AWS SDK for Java – Veuillez consulter AmazonS3EncryptionClient et AmazonS3EncryptionClientV2 dans la Référence d'API AWS SDK for Java.
- AWS SDK for Ruby – Veuillez consulter Aws::S3::Encryption::Client dans la Référence d'API AWS SDK for Ruby.
- AWS SDK for .NET – Veuillez consulter AmazonS3EncryptionClient dans la Référence d'API AWS SDK for .NET.
- AWS SDK for Go – Veuillez consulter s3crypto dans la Référence d'API AWS SDK for Go.

Le client de chiffrement Amazon S3 simplifie le travail d'élaboration des demandes nécessaires adressées à Amazon S3 pour récupérer le message électronique chiffré et adressées à AWS KMS pour déchiffrer la clé de données chiffrée du message, et le travail de déchiffrement du message électronique. Par exemple, pour déchiffrer avec succès la clé de données chiffrée, vous devez transmettre le même contexte de chiffrement que celui transmis par Amazon SES lors de la demande de la clé de données à partir d'AWS KMS (Step 3 dans Présentation du chiffrement Amazon SES à l'aide d'AWS KMS). Le client de chiffrement Amazon S3 gère cela et une grande partie des autres tâches pour vous.

Pour voir un exemple de code qui utilise le client de chiffrement Amazon S3 dans le kit AWS SDK for Java pour effectuer le déchiffrement côté client, veuillez consulter :

- Utilisation d'une clé KMS stockée dans AWS KMS dans le guide du développeur Amazon Simple Storage Service.
- Chiffrement Amazon S3 avec AWS Key Management Service dans le Blog des développeurs AWS.

Comment Amazon Simple Storage Service (Amazon S3) utilise AWS KMS

Amazon Simple Storage Service (Amazon S3) est un service de stockage d'objets qui stocke les données en tant qu'objets dans des compartiments. Les compartiments et les objets qu'ils contiennent sont privés et ne sont accessibles que si vous accordez explicitement des autorisations d'accès.

Amazon S3 s'intègre à AWS Key Management Service (AWS KMS) pour fournir un chiffrement côté serveur des objets Amazon S3. Simple Storage Service (Amazon S3) utilise des clés AWS KMS pour chiffrer vos objets Simple Storage Service (Amazon S3). Les clés de chiffrement qui protègent vos objets ne partent jamaisAWS KMS non chiffré. Cette integration vous permet également de définir des autorisations sur la clé AWS KMS et d'auditer les opérations qui génèrent, chiffrent et déchiffrent les clés de données utilisées pour protéger vos secrets.

Pour réduire le volume d'appels vers Amazon S3AWS KMS, utilisez les clés de compartiment Amazon S3, qui sont protégées par des clés KMS et key-encryption-keys qui sont réutilisées pendant une durée limitée dans Amazon S3. Les clés de compartiment peuvent réduire les coûts pour les demandes AWS KMS jusqu'à 99 %. Vous pouvez configurer une clé de compartiment pour tous les objets dans un compartiment Amazon S3, ou pour un objet spécifique dans un compartiment Amazon S3.

Pour plus d'informations sur la manière dont Simple Storage Service (Amazon S3) utilise AWS KMS, consultez Protection des données à l'aide du chiffrement côté serveur avec des clés KMS (SSE-KMS) dans le Guide de l'utilisateur Simple Storage Service (Amazon S3).

Comment AWS Systems Manager Parameter Store utilise AWS KMS

Avec AWS Systems Manager Parameter Store, vous pouvez créer des paramètres de chaîne sécurisée, qui sont des paramètres dont le nom est en texte brut et la valeur est chiffrée. Parameter Store utilise AWS KMS pour chiffrer et déchiffrer les valeurs des paramètres de chaîne sécurisée.

Avec Parameter Store, vous pouvez créer, stocker et gérer des données sous forme de paramètres assortis de valeurs. Vous pouvez créer un paramètre dans Parameter Store et l'utiliser dans plusieurs applications et services soumis aux stratégies et aux autorisations que vous concevez. Lorsque vous

avez besoin de modifier une valeur de paramètre, au lieu de gérer une modification sujette à erreurs dans diverses sources, vous modifiez une seule instance. Sachant que Parameter Store peut prendre en charge une structure hiérarchique de noms de paramètres, vous pouvez limiter un paramètre à des utilisations spécifiques.

Pour gérer des données sensibles, vous pouvez créer des paramètres de chaîne sécurisée. Parameter Store se sert de AWS KMS keys pour chiffrer les valeurs de paramètres de chaîne sécurisés au moment où vous les créez ou les modifiez. Il utilise également des clés KMS pour déchiffrer les valeurs de paramètres au moment où vous y accédez. Vous pouvez utiliser la Clé gérée par AWS créée par Parameter Store pour votre compte ou spécifier votre propre clé gérée par le client.

> **Important**
>
> Le Parameter Store prend uniquement en charge les clés KMS symétriques. Vous ne pouvez pas utiliser une clé KMS asymétrique pour chiffrer vos paramètres. Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Parameter Store prend en charge deux niveaux de paramètres de chaîne sécurisée : Standard et Avancé. Les paramètres standard, qui ne peuvent pas dépasser 4 096 octets, sont chiffrés et déchiffrés directement sous la clé KMS que vous spécifiez. Pour chiffrer et déchiffrer les paramètres de chaîne sécurisée avancés, Parameter Store utilise le chiffrement d'enveloppe avec le kit AWS Encryption SDK. Vous pouvez convertir un paramètre de chaîne sécurisée standard en un paramètre avancé, mais vous ne pouvez pas convertir un paramètre avancé en paramètre standard. Pour plus d'informations sur la différence entre les paramètres de chaîne sécurisée standard et avancés, veuillez consulter À propos des paramètres avancés de Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Rubriques

- Protection des paramètres standard de chaîne sécurisée
- Protection avancée des paramètres de chaîne sécurisée
- Définition d'autorisations de chiffrement et de déchiffrement des valeurs de paramètres
- Contexte de chiffrement Parameter Store
- Résolution des problèmes de clés KMS dans Parameter Store

Protection des paramètres standard de chaîne sécurisée

Parameter Store n'assure aucune opération cryptographique. Au lieu de cela, il s'appuie sur AWS KMS pour chiffrer et déchiffrer les valeurs de paramètres de chaîne sécurisée. Lorsque vous créez ou modifiez une valeur de paramètre de chaîne sécurisée standard, Parameter Store appelle l'opération AWS KMS Encrypt. Cette opération utilise directement une clé KMS de chiffrement symétrique pour chiffrer la valeur de paramètre au lieu d'utiliser la clé KMS pour générer une clé de données.

Vous pouvez sélectionner la clé KMS dont se sert Parameter Store pour chiffrer la valeur de paramètre. Si vous ne spécifiez pas de clé KMS, Parameter Store utilise la Clé gérée par AWS que Systems Manager crée automatiquement dans votre compte. Cette clé KMS possède l'alias `aws/ssm`.

Pour afficher la clé `aws/ssm` KMS par défaut de votre compte, utilisez l'DescribeKeyopération dans l'AWS KMSAPI. Dans l'exemple suivant, la commande `describe-key` de l'AWS Command Line Interface (AWS CLI) est utilisée avec le nom d'alias `aws/ssm`.

```
aws kms describe-key --key-id alias/aws/ssm
```

Pour créer un paramètre de chaîne sécurisé standard, utilisez l'PutParameteropération de l'API Systems Manager. Omettez le paramètre `Tier` ou spécifiez une valeur `Standard`, qui est la valeur par défaut. Incluez un paramètre `Type` avec la valeur `SecureString`. Pour spécifier une clé KMS, utilisez le paramètre `KeyId`. La valeur par défaut est la Clé gérée par AWS pour votre compte, `aws/ssm`.

Parameter Store appelle alors l'opération `Encrypt` AWS KMS avec la clé KMS et la valeur du paramètre en texte brut. AWS KMS renvoie la valeur chiffrée du paramètre, que Parameter Store stocke avec le nom du paramètre.

Dans l'exemple suivant, la commande Systems Manager put-parameter et son paramètre `--type` sont utilisés dans l'AWS CLI pour créer un paramètre de chaîne sécurisée. Comme la commande omet les paramètres facultatifs `--tier` et `--key-id`, Parameter Store crée un paramètre de chaîne sécurisée standard et le chiffre conformément à la Clé gérée par AWS.

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString
```

Dans l'exemple similaire suivant, le paramètre `--key-id` est utilisé pour spécifier une clé gérée par le client. Cet exemple utilise un ID de clé KMS pour identifier la clé KMS, mais vous pouvez utiliser n'importe quel identifiant de clé KMS valide. Comme la commande omet le paramètre `Tier`

(`--tier`), Parameter Store crée un paramètre de chaîne sécurisée standard, et non un paramètre avancé.

```
aws ssm put-parameter --name param1 --value "secret" --type SecureString --key-id
 1234abcd-12ab-34cd-56ef-1234567890ab
```

Lorsque vous obtenez un paramètre de chaîne sécurisée à partir de Parameter Store, sa valeur est chiffrée. Pour obtenir un paramètre, utilisez l'GetParameter opération dans l'API Systems Manager.

Dans l'exemple suivant, la commande Systems Manager get-parameter est utilisée dans l'AWS CLI pour obtenir le paramètre `MyParameter` à partir de Parameter Store sans déchiffrer sa valeur.

```
$  aws ssm get-parameter --name MyParameter

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

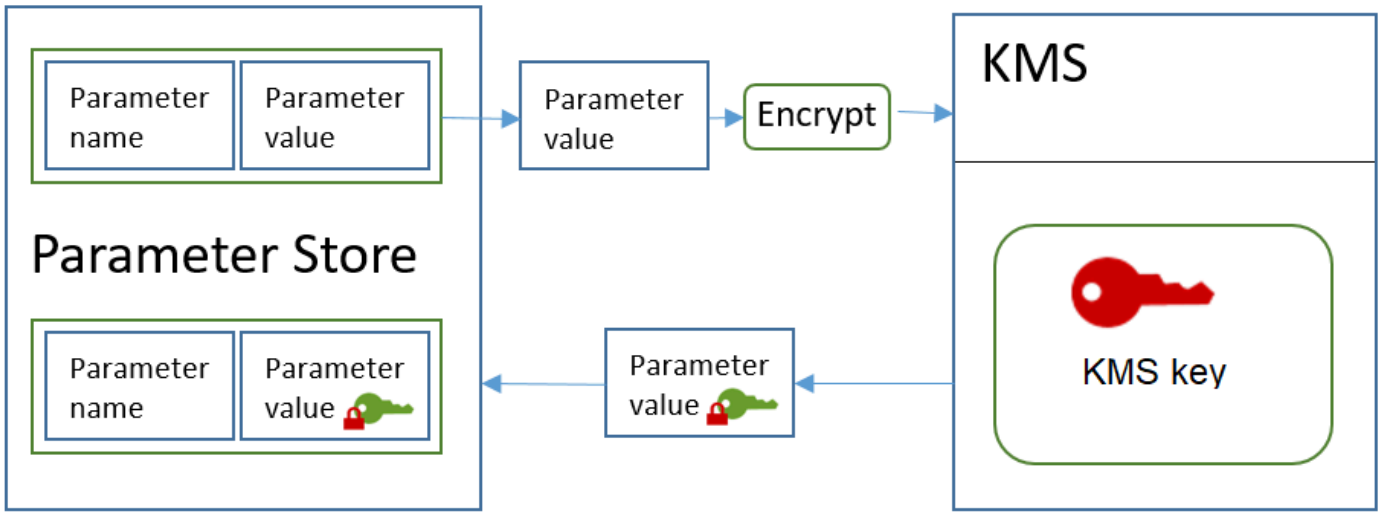
Pour déchiffrer la valeur du paramètre avant de la renvoyer, définissez le paramètre `WithDecryption` de `GetParameter` sur `true`. Lorsque vous utilisez `WithDecryption`, Parameter Store appelle l'opération AWS KMS Decrypt en votre nom pour déchiffrer la valeur du paramètre. En conséquence, la demande `GetParameter` renvoie le paramètre avec une valeur de paramètre en texte brut, comme le montre l'exemple suivant.

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

Le flux de travail suivant montre comment le stockage des paramètres utilise une clé KMS pour chiffrer et déchiffrer un paramètre de chaîne sécurisée standard.

Chiffrer un paramètre standard

1. Lorsque vous utilisez `PutParameter` pour créer un paramètre de chaîne sécurisée, Parameter Store envoie une demande `Encrypt` à AWS KMS. Cette demande inclut la valeur du paramètre en texte brut, la clé KMS que vous avez choisie et le contexte de chiffrement Parameter Store. Pendant la transmission à AWS KMS, la valeur en texte brut du paramètre de chaîne sécurisée est protégée par le protocole TLS (Transport Layer Security).
2. AWS KMS chiffre la valeur du paramètre avec la clé KMS spécifiée et le contexte de chiffrement. Il renvoie le texte chiffré à Parameter Store, qui stocke le nom du paramètre et sa valeur chiffrée.



Déchiffrer un paramètre standard

1. Lorsque vous incluez le paramètre `WithDecryption` dans une demande `GetParameter`, Parameter Store envoie une demande `Decrypt` à AWS KMS avec la valeur chiffrée du paramètre de chaîne sécurisée et le contexte de chiffrement Parameter Store.
2. AWS KMS utilise la même clé KMS et le contexte de chiffrement fourni pour déchiffrer la valeur chiffrée. Il renvoie la valeur du paramètre (déchiffrée) en texte brut à Parameter Store. Pendant la transmission, les données en texte brut sont protégées par TLS.
3. Parameter Store vous renvoie la valeur du paramètre en texte brut dans la réponse `GetParameter`.

Protection avancée des paramètres de chaîne sécurisée

Lorsque vous utilisez `PutParameter` pour créer un paramètre de chaîne sécurisée avancé, Parameter Store utilise l'enveloppe de chiffrement avec le kit AWS Encryption SDK et une AWS KMS key de chiffrement symétrique pour protéger la valeur du paramètre. Chaque valeur de paramètre avancé est chiffrée sous une clé de données unique, et la clé de données est chiffrée sous une clé KMS. Vous pouvez utiliser la Clé gérée par AWS pour le compte (`aws/ssm`) ou n'importe quelle clé gérée par le client.

La bibliothèque AWS Encryption SDK est une bibliothèque côté client et open source qui vous permet de chiffrer et de déchiffrer les données à l'aide des normes et des bonnes pratiques. Elle est prise en charge sur plusieurs plateformes et dans plusieurs langages de programmation, y compris une interface de ligne de commande. Vous pouvez consulter le code source et contribuer à son développement dans GitHub.

Pour chaque valeur de paramètre de chaîne sécurisée, Parameter Store appelle le AWS Encryption SDK pour chiffrer la valeur du paramètre à l'aide d'une clé de données unique qui AWS KMS génère (GenerateDataKey). Le kit AWS Encryption SDK renvoie à Parameter Store un message chiffré qui inclut la valeur du paramètre chiffrée et une copie chiffrée de la clé de données unique. Parameter Store stocke l'intégralité du message chiffré dans la valeur du paramètre de chaîne sécurisée. Ensuite, lorsque vous obtenez une valeur de paramètre de chaîne sécurisée avancé, Parameter Store utilise le kit AWS Encryption SDK pour déchiffrer la valeur du paramètre. Ceci nécessite un appel à AWS KMS pour déchiffrer la clé de données chiffrée.

Pour créer un paramètre de chaîne sécurisé avancé, utilisez l'PutParameteropération de l'API Systems Manager. Définissez la valeur du paramètre `Tier` sur `Advanced`. Incluez un paramètre `Type` avec la valeur `SecureString`. Pour spécifier une clé KMS, utilisez le paramètre `KeyId`. La valeur par défaut est la Clé gérée par AWS pour votre compte, `aws/ssm`.

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString --
tier Advanced
```

Dans l'exemple similaire suivant, le paramètre `--key-id` est utilisé pour spécifier une clé gérée par le client. Cet exemple utilise l'Amazon Resource Name (ARN) de la clé KMS, mais vous pouvez utiliser tout identifiant de clé KMS valide.

```
aws ssm put-parameter --name MyParameter --value "secret_value"
 --type SecureString --tier Advanced --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Lorsque vous obtenez un paramètre de chaîne sécurisée à partir de Parameter Store, sa valeur est le message chiffré que le kit AWS Encryption SDK a renvoyé. Pour obtenir un paramètre, utilisez l'GetParameter opération dans l'API Systems Manager.

L'exemple suivant utilise l'opération `GetParameter` Systems Manager pour obtenir le paramètre `MyParameter` à partir de Parameter Store sans déchiffrement de sa valeur.

```
$  aws ssm get-parameter --name MyParameter

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

Pour déchiffrer la valeur du paramètre avant de la renvoyer, définissez le paramètre `WithDecryption` de `GetParameter` sur `true`. Lorsque vous utilisez `WithDecryption`, Parameter Store appelle l'opération AWS KMS Decrypt en votre nom pour déchiffrer la valeur du paramètre. En conséquence, la demande `GetParameter` renvoie le paramètre avec une valeur de paramètre en texte brut, comme le montre l'exemple suivant.

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

Vous ne pouvez pas convertir un paramètre de chaîne sécurisée avancé en un paramètre standard, mais vous pouvez convertir un paramètre de chaîne sécurisée standard en paramètre avancé. Pour convertir un paramètre de chaîne sécurisée standard en chaîne sécurisée avancée, utilisez l'opération `PutParameter` avec le paramètre `Overwrite`. Le `Type` doit être `SecureString` et la valeur `Tier` doit être `Advanced`. Le paramètre `KeyId`, qui identifie une clé gérée par le client, est facultatif. Si vous l'omettez, Parameter Store utilise la Clé gérée par AWS pour le compte. Vous

pouvez spécifier n'importe quelle clé KMS que le principal est autorisé à utiliser, même si vous avez utilisé une autre clé KMS pour chiffrer le paramètre standard.

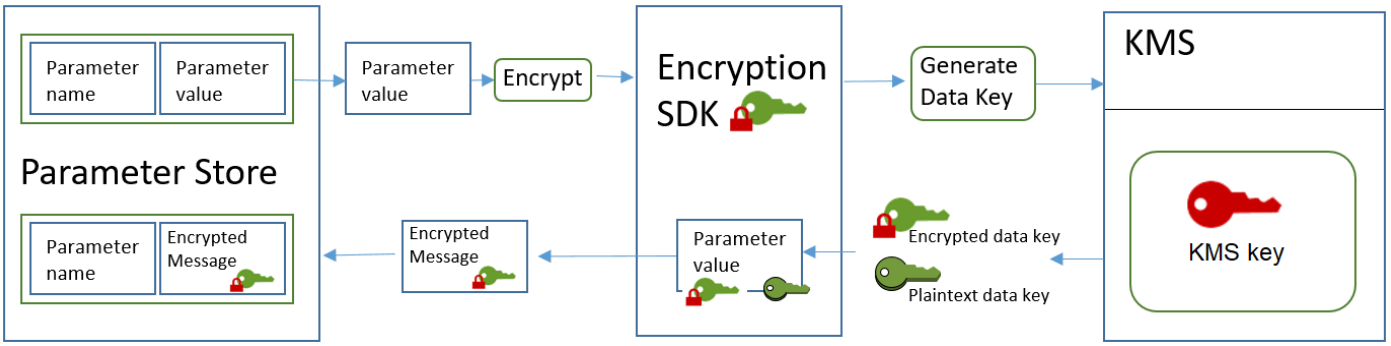
Lorsque vous utilisez le paramètre `Overwrite`, Parameter Store utilise le kit AWS Encryption SDK pour chiffrer la valeur du paramètre. Ensuite, il stocke le nouveau message chiffré dans Parameter Store.

```
$ aws ssm put-parameter --name myStdParameter --value "secret_value"  --type
 SecureString --tier Advanced --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --overwrite
```

Le flux de travail suivant montre comment Parameter Store utilise une clé KMS pour chiffrer et déchiffrer un paramètre de chaîne sécurisée avancé.

Chiffrer un paramètre avancé

1. Lorsque vous utilisez `PutParameter` pour créer un paramètre de chaîne sécurisée avancé, Parameter Store utilise le kit AWS Encryption SDK et AWS KMS pour chiffrer la valeur du paramètre. Parameter Store appelle le kit AWS Encryption SDK avec la valeur du paramètre, la clé KMS que vous avez spécifiée et le contexte de chiffrement Parameter Store.
2. AWS Encryption SDKEnvoie une GenerateDataKeydemande AWS KMS avec l'identifiant de la clé KMS que vous avez spécifiée et le contexte de chiffrement du Parameter Store. AWS KMSrenvoie deux copies de la clé de données unique : l'une en texte brut et l'autre cryptée sous la clé KMS. (Le contexte de chiffrement est utilisé lors du chiffrement de la clé de données.)
3. Le kit AWS Encryption SDK utilise la clé de données en texte brut pour chiffrer les données. Elle renvoie un message chiffré qui inclut la valeur du paramètre chiffré, la clé de données chiffrée, et d'autres données, y compris le contexte de chiffrement Parameter Store.
4. Parameter Store stocke le message chiffré en tant que valeur de paramètre.



Déchiffrer un paramètre avancé

1. Vous pouvez inclure le paramètre `WithDecryption` dans une demande `GetParameter` pour obtenir un paramètre de chaîne sécurisée avancé. Lorsque vous le faites, Parameter Store transmet le message chiffré de la valeur de paramètre à une méthode de déchiffrement du kit AWS Encryption SDK.
2. Le kit AWS Encryption SDK appelle l'opération AWS KMS Decrypt. Il transmet la clé de données chiffrée et le contexte de chiffrement Parameter Store depuis le message chiffré.
3. AWS KMS utilise la clé KMS et le contexte de chiffrement Parameter Store pour déchiffrer la clé de données chiffrée. Ensuite, il renvoie la clé de données en texte brut (déchiffrée) au kit AWS Encryption SDK.
4. Le kit AWS Encryption SDK utilise la clé de données en texte brut pour déchiffrer la valeur du paramètre. Il renvoie la valeur du paramètre en texte brut à Parameter Store.
5. Parameter Store vérifie le contexte de chiffrement et renvoie la valeur du paramètre en texte brut dans la réponse `GetParameter`.

Définition d'autorisations de chiffrement et de déchiffrement des valeurs de paramètres

Pour chiffrer une valeur de paramètre de chaîne sécurisée standard, l'utilisateur a besoin d'une autorisation `kms:Encrypt`. Pour chiffrer une valeur de paramètre de chaîne sécurisée avancée, l'utilisateur a besoin d'une autorisation `kms:GenerateDataKey`. Pour déchiffrer n'importe quel type de valeur de paramètre de chaîne sécurisée, l'utilisateur a besoin d'une autorisation `kms:Decrypt`.

Vous pouvez utiliser des stratégies IAM pour accorder ou refuser l'autorisation à un utilisateur d'appeler les opérations Systems Manager `PutParameter` et `GetParameter`.

De plus, si vous utilisez les clés gérées par le client pour chiffrer les valeurs de vos paramètres de chaîne sécurisée, vous pouvez utiliser des stratégies IAM et des stratégies de clé pour gérer les autorisations de chiffrement et de déchiffrement. Cependant, vous ne pouvez pas établir de stratégies de contrôle d'accès pour la clé KMS `aws/ssm` par défaut. Pour en savoir plus sur le contrôle d'accès aux clés gérées par le client, veuillez consulter Authentification et contrôle d'accès pour AWS KMS.

L'exemple suivant montre une stratégie IAM conçue pour les paramètres de chaîne de sécurité standard. Elle permet à l'utilisateur d'appeler l'opération `PutParameter` Systems Manager sur tous les paramètres du chemin `FinancialParameters`. Cette stratégie permet également à l'utilisateur d'appeler l'opération AWS KMS `Encrypt` sur la clé d'exemple gérée par le client.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/
FinancialParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

L'exemple suivant illustre une stratégie IAM conçue pour les paramètres de chaîne de sécurité avancés. Elle permet à l'utilisateur d'appeler l'opération `PutParameter` Systems Manager sur tous les paramètres du chemin `ReservedParameters`. Cette stratégie permet également à l'utilisateur d'appeler l'opération AWS KMS `GenerateDataKey` sur la clé d'exemple gérée par le client.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/
ReservedParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
```

```
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Le dernier exemple montre également une stratégie IAM qui peut être utilisée pour les paramètres de chaîne de sécurité avancés ou standard. Elle permet à l'utilisateur d'appeler les opérations `GetParameter` Systems Manager (et les opérations associées) sur tous les paramètres du chemin `ITParameters`. Cette stratégie permet également à l'utilisateur d'appeler l'opération AWS KMS `Decrypt` sur la clé d'exemple gérée par le client.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/ITParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Contexte de chiffrement Parameter Store

Un contexte de chiffrement est un ensemble de paires clé-valeur qui contiennent des données non secrètes arbitraires. Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de données, AWS KMS lie de manière chiffrée le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez transmettre le même contexte de chiffrement.

Vous pouvez également utiliser le contexte de chiffrement pour identifier une opération de chiffrement dans des enregistrements d'audit et des journaux. Le contexte de chiffrement s'affiche en texte brut dans les journaux, tels que les journaux AWS CloudTrail.

Le kit AWS Encryption SDK accepte aussi un contexte de chiffrement, même s'il peut le gérer différemment. Parameter Store fournit le contexte de chiffrement à la méthode de chiffrement. Le kit AWS Encryption SDK lie cryptographiquement le contexte de chiffrement aux données chiffrées. Il inclut aussi le contexte de chiffrement en texte brut dans l'en-tête du message chiffré qu'il renvoie. Toutefois, contrairement à AWS KMS, les méthodes de déchiffrement AWS Encryption SDK ne prennent pas un contexte de chiffrement comme entrée. Au lieu de cela, lorsqu'il déchiffre des données, le kit AWS Encryption SDK obtient le contexte de chiffrement à partir du message chiffré. Parameter Store vérifie que le contexte de chiffrement inclut la valeur qu'il attend avant de vous renvoyer la valeur du paramètre en texte brut.

Parameter Store utilise le contexte de chiffrement suivant dans ses opérations cryptographiques :

- Clé : `PARAMETER_ARN`
- Valeur : Amazon Resource Name (ARN) du paramètre chiffré.

Le format du contexte de chiffrement est le suivant :

```
"PARAMETER_ARN":"arn:aws:ssm:<REGION_NAME>:<ACCOUNT_ID>:parameter/<parameter-name>"
```

Par exemple, Parameter Store inclut ce contexte de chiffrement dans les appels à chiffrer et déchiffrer le paramètre `MyParameter` dans un exemple de compte et de région Compte AWS.

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
```

Si le paramètre se trouve dans le chemin hiérarchique de Parameter Store, le chemin et le nom sont inclus dans le contexte de chiffrement. Par exemple, ce contexte de chiffrement est utilisé lors du chiffrement et du déchiffrement du paramètre `MyParameter` dans le chemin `/ReadableParameters` d'un exemple de compte et de région Compte AWS.

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/ReadableParameters/
MyParameter"
```

Vous pouvez déchiffrer une valeur chiffrée de paramètre de chaîne sécurisée en appelant l'opération AWS KMS `Decrypt` avec le contexte de chiffrement adéquat et la valeur chiffrée du paramètre

renvoyée par l'opération Systems Manager `GetParameter`. Cependant, nous vous encourageons à déchiffrer les valeurs de paramètre Parameter Store à l'aide de l'opération `GetParameter` avec le paramètre `WithDecryption`.

Vous pouvez également inclure le contexte de chiffrement dans une stratégie IAM. Par exemple, vous pouvez autoriser un utilisateur à déchiffrer la valeur d'un seul et même paramètre ou les valeurs d'un ensemble de paramètres.

L'exemple suivant d'instruction de stratégie IAM permet à l'utilisateur d'obtenir la valeur du paramètre `MyParameter` et de déchiffrer sa valeur à l'aide de la clé KMS spécifiée. Toutefois, les autorisations s'appliquent uniquement lorsque le contexte de chiffrement correspond à la chaîne spécifiée. Ces autorisations ne s'appliquent à aucun autre paramètre ou à aucune autre clé KMS, et l'appel de `GetParameter` échoue si le contexte de chiffrement ne correspond pas à la chaîne.

Avant d'utiliser une déclaration de stratégie comme celle-ci, remplacez les exemples d'ARN avec des valeurs valides.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:PARAMETER_ARN":"arn:aws:ssm:us-
west-2:111122223333:parameter/MyParameter"
                }
            }
        }
    ]
```

```
}
```

Résolution des problèmes de clés KMS dans Parameter Store

Pour effectuer une opération sur un paramètre de chaîne sécurisée, Parameter Store doit pouvoir utiliser la clé KMS AWS KMS que vous spécifiez pour l'opération en question. La plupart des échecs de Parameter Store liés aux clés KMS sont imputables aux problèmes suivants :

- Les informations d'identification utilisées par une application ne disposent pas des autorisations nécessaires pour effectuer l'action spécifiée sur la clé KMS.

  Pour corriger cette erreur, exécutez l'application avec d'autres informations d'identification ou corrigez la politique IAM ou la politique de clé qui empêche l'opération. Pour obtenir de l'aide concernant les stratégies IAM et de clé AWS KMS, consultez Authentification et contrôle d'accès pour AWS KMS.
- La clé KMS est introuvable.

  Cela se produit généralement lorsque vous utilisez un identificateur incorrect pour la clé KMS. Procurez-vous les identificateurs corrects pour la clé KMS et tentez à nouveau la commande.
- La clé KMS n'est pas activée. Dans ce cas, Parameter Store renvoie une InvalidKeyIdexception avec un message d'erreur détaillé provenant deAWS KMS. Si la clé KMS est à l'état `Disabled`, activez-la. Si l'état est `Pending Import`, suivez la procédure d'importation. Si la clé est à l'état `Pending Deletion`, annulez la suppression de la clé ou utilisez une autre clé KMS.

  Pour obtenir l'état de la clé d'une clé KMS dans la console AWS KMS, sur la page Clés gérées par le client ou Clés gérées par AWS, veuillez consulter la colonne Status (Statut). Pour utiliser l'AWS KMSAPI afin de connaître l'état d'une clé KMS, utilisez l'DescribeKeyopération.

Comment Amazon WorkMail utilise AWS KMS

Cette rubrique explique comment WorkMail Amazon AWS KMS crypte les e-mails.

Rubriques

- WorkMail Présentation d'Amazon
- WorkMail Chiffrement Amazon
- Autoriser l'utilisation de la clé KMS
- Contexte WorkMail de chiffrement Amazon

- Surveillance de WorkMail l'interaction d'Amazon avec AWS KMS

WorkMail Présentation d'Amazon

Amazon WorkMail est un service de messagerie et de calendrier professionnel sécurisé et géré qui prend en charge les clients de messagerie de bureau et mobiles existants. Vous pouvez créer une WorkMail organisation Amazon et lui attribuer un ou plusieurs domaines de messagerie dont vous êtes le propriétaire. Ensuite, vous pouvez créer des boîtes de réception pour les e-mails des utilisateurs et des groupes de distribution de l'organisation.

Amazon chiffre de WorkMail manière transparente tous les messages contenus dans les boîtes aux lettres de toutes les WorkMail organisations Amazon avant qu'ils ne soient écrits sur le disque et les déchiffre de manière transparente lorsque les utilisateurs y accèdent. Il n'y a aucune option pour désactiver le chiffrement. Pour protéger les clés de chiffrement qui protègent les messages, Amazon WorkMail est intégré à AWS Key Management Service (AWS KMS).

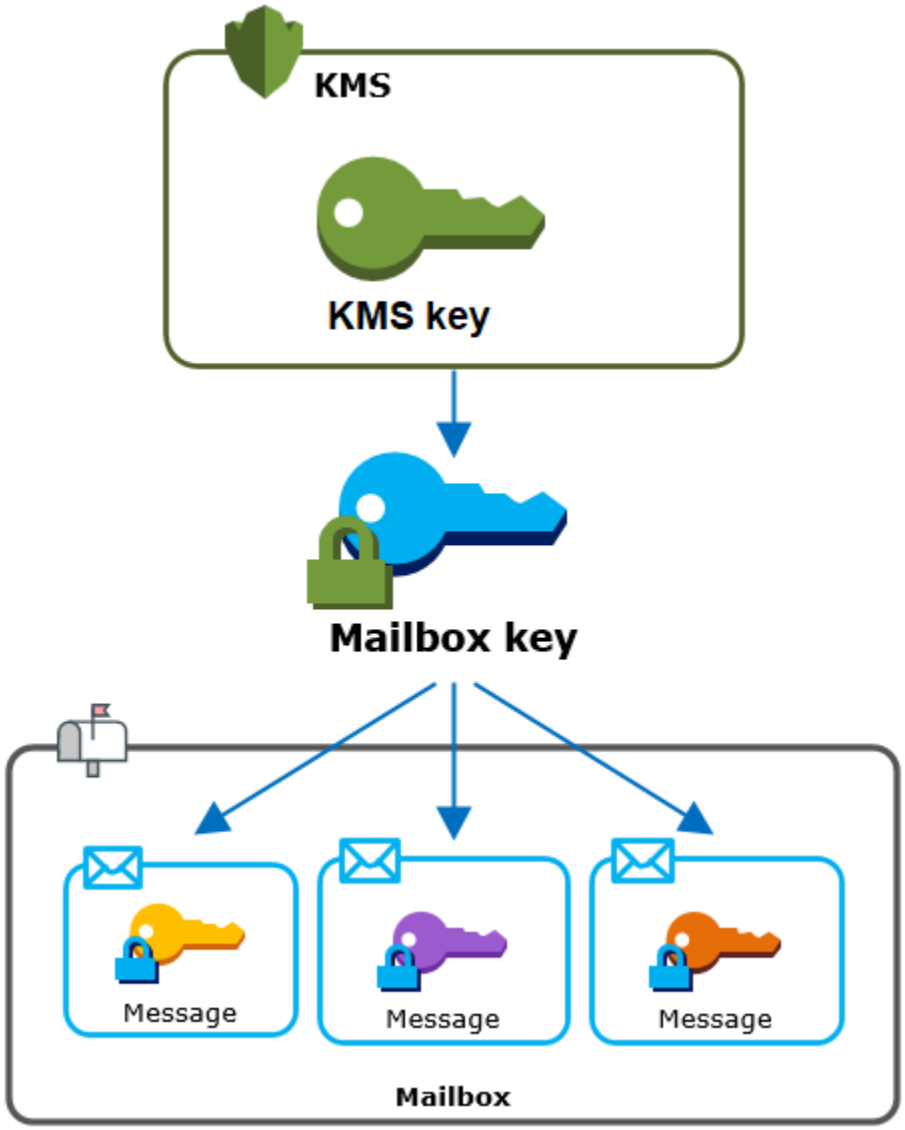
Amazon propose WorkMail également une option permettant aux utilisateurs d'envoyer des e-mails signés ou chiffrés. Cette fonctionnalité de chiffrement n'utilise pas AWS KMS.

WorkMail Chiffrement Amazon

Dans Amazon WorkMail, chaque organisation peut contenir plusieurs boîtes aux lettres, une pour chaque utilisateur de l'organisation. Tous les messages, y compris les e-mails et les éléments de calendrier, sont stockés dans la boîte de réception de l'utilisateur.

Pour protéger le contenu des boîtes aux lettres de vos WorkMail organisations Amazon, Amazon WorkMail chiffre tous les messages des boîtes aux lettres avant qu'ils ne soient écrits sur le disque. Aucune des informations fournies par le client n'est stockée en texte brut.

Chaque message est chiffré sous une clé de chiffrement de données unique. La clé du message est protégée par une clé de boîte de réception, qui est une clé de chiffrement unique utilisée uniquement pour cette boîte de réception. La clé de la boîte de réception est chiffrée sous une AWS KMS key pour l'organisation qui ne laisse jamais AWS KMS non chiffré. Le schéma suivant illustre la relation entre les messages chiffrés, les clés des messages chiffrés, la clé de la boîte de réception chiffrée et la clé KMS de l'organisation dans AWS KMS.



Une clé KMS pour l'organisation

Lorsque vous créez une WorkMail organisation Amazon, vous pouvez en sélectionner une AWS KMS key pour l'organisation. Cette clé KMS protège toutes les clés de boîte de réception de cette organisation.

Si vous utilisez la procédure de configuration rapide pour créer votre organisation, Amazon WorkMail utilise le Clé gérée par AWSfor Amazon WorkMail (`aws/workmail`) dans votreCompte AWS.
Si vous utilisez la configuration standard, vous pouvez sélectionner la clé Clé gérée par AWS pour Amazon WorkMail ou une clé gérée par le client que vous possédez et gérez. Vous pouvez sélectionner la même clé KMS ou une autre clé KMS pour chacune de vos organisations, mais vous ne pouvez pas modifier la clé KMS une fois que vous l'avez sélectionnée.

> **Important**
>
> Amazon WorkMail prend uniquement en charge les clés KMS de chiffrement symétriques. Vous ne pouvez pas utiliser de clé KMS asymétrique pour chiffrer des données sur Amazon. WorkMail Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, veuillez consulter Identification des clés KMS asymétriques.

Pour rechercher la clé KMS de votre organisation, utilisez l'entrée de journal AWS CloudTrail qui enregistre les appels vers AWS KMS.

Clé de chiffrement unique pour chaque boîte de réception

Lorsque vous créez une nouvelle boîte aux lettres, Amazon WorkMail génère une clé de chiffrement symétrique AES (Advanced Encryption Standard) 256 bits unique pour la boîte aux lettres, connue sous le nom de clé de boîte aux lettres, en dehors de. AWS KMS Amazon WorkMail utilise la clé de boîte aux lettres pour protéger les clés de chiffrement de chaque message contenu dans la boîte aux lettres.

Pour protéger la clé de boîte aux lettres, Amazon WorkMail appelle AWS KMS pour chiffrer la clé de boîte aux lettres sous la clé KMS de l'organisation. Ensuite, il stocke la clé de la boîte de réception chiffrée dans la boîte de réception des métadonnées.

> **Note**
>
> Amazon WorkMail utilise une clé de chiffrement de boîte aux lettres symétrique pour protéger les clés de message. Auparavant, Amazon WorkMail protégeait chaque boîte aux lettres à l'aide d'une paire de clés asymétrique. Il utilise la clé publique pour chiffrer chaque clé de message et la clé privée pour la déchiffrer. La clé privée de la boîte de réception a été protégée par la clé KMS de l'organisation. Les boîtes aux lettres existantes peuvent toujours utiliser une paire de clés de boîte de réception asymétrique. Cette modification n'a pas d'incidence sur la sécurité de la boîte de réception ou de ses messages.

Clé de chiffrement unique pour chaque message

Lorsqu'un message est ajouté à la boîte aux lettres, Amazon WorkMail génère une clé de chiffrement symétrique AES 256 bits unique pour le message en dehors de. AWS KMS Il utilise cette clé de message pour chiffrer le message. Amazon WorkMail chiffre la clé du message sous la clé de la boîte

aux lettres et stocke la clé de message chiffrée avec le message. Ensuite, il chiffre la clé de la boîte de réception sous la clé KMS de l'organisation.

Création d'une boîte de réception

Lorsqu'Amazon WorkMail crée une nouvelle boîte aux lettres, il utilise le processus suivant pour préparer la boîte aux lettres afin qu'elle contienne des messages chiffrés.

- Amazon WorkMail génère une clé de chiffrement symétrique AES 256 bits unique pour la boîte aux lettres située à l'extérieur. AWS KMS
- Amazon WorkMail lance l'opération AWS KMS Encrypt. Il transmet la clé de la boîte de réception et l'identificateur de AWS KMS key de l'organisation. AWS KMS renvoie un texte chiffré de la clé de la boîte de réception chiffrée sous la clé KMS.
- Amazon WorkMail stocke la clé cryptée de la boîte aux lettres avec les métadonnées de la boîte aux lettres.

Chiffrement d'un message de boîte de réception

Pour chiffrer un message, Amazon WorkMail utilise le processus suivant.

1. Amazon WorkMail génère une clé symétrique AES 256 bits unique pour le message. Il utilise la clé de données en texte brut et l'algorithme AES (Advanced Encryption Standard) pour chiffrer la valeur du secret en dehors d'AWS KMS.
2. Pour protéger la clé de message située sous la clé de boîte aux lettres, Amazon WorkMail doit déchiffrer la clé de boîte aux lettres, qui est toujours stockée sous sa forme cryptée.

   Amazon WorkMail lance l'opération AWS KMS Decrypt et transmet la clé cryptée de la boîte aux lettres. AWS KMSutilise la clé KMS pour que l'organisation déchiffre la clé de boîte aux lettres et renvoie la clé de boîte aux lettres en texte clair à Amazon. WorkMail
3. Amazon WorkMail utilise la clé de boîte aux lettres en texte brut et l'algorithme Advanced Encryption Standard (AES) pour chiffrer la clé du message en dehors de. AWS KMS
4. Amazon WorkMail stocke la clé du message chiffré dans les métadonnées du message chiffré afin qu'elle soit disponible pour le déchiffrer.

Déchiffrement d'un message de la boîte de réception

Pour déchiffrer un message, Amazon WorkMail utilise le processus suivant.

1. Amazon WorkMail lance l'opération AWS KMS Decrypt et transmet la clé cryptée de la boîte aux lettres. AWS KMSutilise la clé KMS pour que l'organisation déchiffre la clé de boîte aux lettres et renvoie la clé de boîte aux lettres en texte clair à Amazon. WorkMail
2. Amazon WorkMail utilise la clé de boîte aux lettres en texte brut et l'algorithme Advanced Encryption Standard (AES) pour déchiffrer la clé de message chiffrée en dehors de. AWS KMS
3. Amazon WorkMail utilise la clé du message en texte brut pour déchiffrer le message chiffré.

Mise en cache des clés de boîte de réception

Pour améliorer les performances et minimiser les appelsAWS KMS, Amazon met en WorkMail cache chaque clé de boîte aux lettres en texte brut pour chaque client localement pendant une minute maximum. À la fin de la période de mise en cache, la clé de la boîte de réception est supprimée. Si la clé de boîte aux lettres de ce client est requise pendant la période de mise en cache, Amazon WorkMail peut l'obtenir depuis le cache au lieu d'appelerAWS KMS. La clé de la boîte de réception est protégée dans le cache et n'est jamais écrit sur le disque en texte brut.

Autoriser l'utilisation de la clé KMS

Lorsqu'Amazon WorkMail utilise une AWS KMS key opération cryptographique, il agit pour le compte de l'administrateur de la boîte aux lettres.

Pour utiliser la AWS KMS key d'un secret en votre nom, l'utilisateur doit disposer des autorisations suivantes. Vous pouvez spécifier ces autorisations requises dans une politique IAM ou dans une politique de clé.

- `kms:Encrypt`
- `kms:Decrypt`
- `kms:CreateGrant`

Pour autoriser l'utilisation de la clé KMS uniquement pour les demandes provenant d'Amazon WorkMail, vous pouvez utiliser la clé de ViaService condition kms : avec la `workmail.<region>.amazonaws.com` valeur.

Vous pouvez également utiliser les clés ou les valeurs du contexte de chiffrement comme condition d'utilisation de la clé KMS pour les opérations cryptographiques. Par exemple, vous pouvez utiliser un opérateur de condition de chaîne dans un document de politique IAM ou de clé, ou utiliser une contrainte d'octroi dans un octroi.

Politique de la Clé gérée par AWS

La politique clé Clé gérée par AWS pour Amazon WorkMail autorise les utilisateurs à utiliser la clé KMS pour des opérations spécifiques uniquement lorsqu'Amazon WorkMail fait la demande au nom de l'utilisateur. La politique de clé n'autorise pas les utilisateurs à utiliser la clé KMS directement.

Cette politique de clé, comme les politiques de toutes les Clés gérées par AWS, est établie par le service. Vous ne pouvez pas modifier la politique de clé, mais vous pouvez l'afficher à tout moment. Pour plus de détails, veuillez consulter Affichage d'une politique de clé.

Les instructions de politique de la politique de clé ont l'effet suivant :

- Autorisez les utilisateurs du compte et de la région à utiliser la clé KMS pour les opérations cryptographiques et pour créer des autorisations, mais uniquement lorsque la demande provient d'Amazon en leur WorkMail nom. La clé de condition `kms:ViaService` applique cette restriction.
- Autorise le Compte AWS à créer des politiques IAM qui permettent aux utilisateurs d'afficher les propriétés de clé KMS et de révoquer des octrois.

Voici une politique clé à titre d'exemple Clé gérée par AWS pour Amazon WorkMail.

```
{
  "Version" : "2012-10-17",
  "Id" : "auto-workmail-1",
  "Statement" : [ {
    "Sid" : "Allow access through WorkMail for all principals in the account that are
 authorized to use WorkMail",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [ "kms:Decrypt", "kms:CreateGrant", "kms:ReEncrypt*", "kms:DescribeKey",
 "kms:Encrypt" ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "workmail.us-east-1.amazonaws.com",
        "kms:CallerAccount" : "111122223333"
      }
    }
  }, {
    "Sid" : "Allow direct access to key metadata to the account",
```

```
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : [ "kms:Describe*", "kms:List*", "kms:Get*", "kms:RevokeGrant" ],
    "Resource" : "*"
  } ]
}
```

Utiliser des subventions pour autoriser Amazon WorkMail

Outre les politiques clés, Amazon WorkMail utilise des subventions pour ajouter des autorisations à la clé KMS pour chaque organisation. Pour consulter les autorisations associées à la clé KMS de votre compte, utilisez l'ListGrantsopération.

Amazon WorkMail utilise des autorisations pour ajouter les autorisations suivantes à la clé KMS de l'organisation.

- Ajoutez l'`kms:Encrypt`autorisation permettant à Amazon de chiffrer WorkMail la clé de la boîte aux lettres.
- Ajoutez l'`kms:Decrypt`autorisation permettant à Amazon d' WorkMail utiliser la clé KMS pour déchiffrer la clé de boîte aux lettres. Amazon WorkMail exige cette autorisation dans le cadre d'une autorisation, car la demande de lecture des messages de boîte aux lettres utilise le contexte de sécurité de l'utilisateur qui lit le message. La demande n'utilise pas les informations d'identification de la Compte AWS. Amazon WorkMail crée cette subvention lorsque vous sélectionnez une clé KMS pour l'organisation.

Pour créer les subventions, Amazon WorkMail appelle au CreateGrantnom de l'utilisateur qui a créé l'organisation. L'autorisation de créer l'octroi provient de la politique de clé. Cette politique permet aux utilisateurs du compte d'appeler `CreateGrant` la clé KMS de l'organisation lorsqu'Amazon WorkMail fait la demande au nom d'un utilisateur autorisé.

La politique de clé autorise également la racine du compte à révoquer l'octroi sur la Clé gérée par AWS. Toutefois, si vous révoquez l'autorisation, Amazon WorkMail ne pourra pas déchiffrer les données chiffrées de vos boîtes aux lettres.

Contexte WorkMail de chiffrement Amazon

Un contexte de chiffrement est un ensemble de paires clé-valeur contenant les données non secrètes arbitraires. Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de

données, AWS KMS lie de manière chiffrée le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez transmettre le même contexte de chiffrement.

Amazon WorkMail utilise le même format de contexte de chiffrement dans toutes les opérations AWS KMS cryptographiques. Vous pouvez utiliser le contexte de chiffrement pour identifier une opération de chiffrement dans les enregistrements d'audit et les journaux, tels qu' AWS CloudTrail, et en tant que condition pour l'autorisation dans les politiques et les octrois.

Dans ses demandes Encrypt and Decrypt à, AWS KMS Amazon WorkMail utilise un contexte de chiffrement dans lequel la clé `aws:workmail:arn` et la valeur sont le nom de ressource Amazon (ARN) de l'organisation.

```
"aws:workmail:arn":"arn:aws:workmail:region:account ID:organization/organization ID"
```

Par exemple, le contexte de chiffrement suivant inclut un exemple d'ARN d'organisation dans la région USA Est (Ohio) (`us-east-2`).

```
"aws:workmail:arn":"arn:aws:workmail:us-east-2:111122223333:organization/
m-68755160c4cb4e29a2b2f8fb58f359d7"
```

Surveillance de WorkMail l'interaction d'Amazon avec AWS KMS

Vous pouvez utiliser AWS CloudTrail Amazon CloudWatch Logs pour suivre les demandes qu'Amazon WorkMail envoie AWS KMS en votre nom.

Encrypt

Lorsque vous créez une nouvelle boîte aux lettres, Amazon WorkMail génère une clé de boîte aux lettres et appelle AWS KMS pour chiffrer la clé de boîte aux lettres. Amazon WorkMail envoie une demande de chiffrement contenant la clé de boîte aux lettres en texte brut et un identifiant pour la clé KMS de l'organisation Amazon WorkMail . AWS KMS

L'événement qui enregistre l'opération `Encrypt` est similaire à l'exemple d'événement suivant. L'utilisateur est le WorkMail service Amazon. Les paramètres incluent l'ID de clé KMS (`keyId`) et le contexte de chiffrement pour l' WorkMail organisation Amazon. Amazon transmet WorkMail également la clé de la boîte aux lettres, mais celle-ci n'est pas enregistrée dans le CloudTrail journal.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
```

```
        "type": "AWSService",
        "invokedBy": "workmail.eu-west-1.amazonaws.com"
    },
    "eventTime": "2019-02-19T10:01:09Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "workmail.eu-west-1.amazonaws.com",
    "userAgent": "workmail.eu-west-1.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:workmail:arn": "arn:aws:workmail:eu-west-1:111122223333:organization/
m-c6981ff7642446fa8772ba99c690e455"
        },
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    },
    "responseElements": null,
    "requestID": "76e96b96-7e24-4faf-a2d6-08ded2eaf63c",
    "eventID": "d5a59c18-128a-4082-aa5b-729f7734626a",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "sharedEventID": "d08e60f1-097e-4a00-b7e9-10bc3872d50c"
}
```


Decrypt

Lorsque vous ajoutez, consultez ou supprimez un message de boîte aux lettres, Amazon WorkMail demande AWS KMS à déchiffrer la clé de la boîte aux lettres. Amazon WorkMail envoie une demande de déchiffrement à l'AWS KMSaide de la clé de boîte aux lettres cryptée et d'un identifiant pour la clé KMS de l' WorkMailorganisation Amazon.

L'événement qui enregistre l'opération `Decrypt` est similaire à l'exemple d'événement suivant. L'utilisateur est le WorkMail service Amazon. Les paramètres incluent la clé de boîte aux lettres

cryptée (sous forme de blob de texte chiffré), qui n'est pas enregistrée dans le journal, et le contexte de chiffrement pour l'organisation Amazon. WorkMail AWS KMSdéduit l'ID de la clé KMS à partir du texte chiffré.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "workmail.eu-west-1.amazonaws.com"
    },
    "eventTime": "2019-02-20T11:51:10Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "workmail.eu-west-1.amazonaws.com",
    "userAgent": "workmail.eu-west-1.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:workmail:arn": "arn:aws:workmail:eu-west-1:111122223333:organization/
m-c6981ff7642446fa8772ba99c690e455"
        }
    },
    "responseElements": null,
    "requestID": "4a32dda1-34d9-4100-9718-674b8e0782c9",
    "eventID": "ea9fd966-98e9-4b7b-b377-6e5a397a71de",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "sharedEventID": "241e1e5b-ff64-427a-a5b3-7949164d0214"
}
```

Comment WorkSpaces utilise AWS KMS

Vous pouvez l'utiliser WorkSpacespour fournir un poste de travail basé sur le cloud (a WorkSpace) pour chacun de vos utilisateurs finaux. Lorsque vous en lancez un nouveau WorkSpace, vous pouvez choisir de chiffrer ses volumes et décider lequel utiliser AWS KMS keypour le chiffrement. Vous pouvez choisir la clé Clé gérée par AWSfor WorkSpaces (aws/espaces de travail) ou une clé symétrique gérée par le client.

> **Important**
>
> WorkSpaces prend uniquement en charge les clés KMS de chiffrement symétriques. Vous ne pouvez pas utiliser de clé KMS asymétrique pour chiffrer les volumes d'un. WorkSpaces Pour obtenir de l'aide sur la détermination de la symétrie ou de l'asymétrie d'une clé KMS, consultez Identification des clés KMS asymétriques.

Pour plus d'informations sur la création à l' WorkSpaces aide de volumes chiffrés, consultez la section Encrypt a WorkSpace du guide d' WorkSpaces administration Amazon.

Rubriques

- Vue d'ensemble du WorkSpaces chiffrement à l'aide de AWS KMS
- WorkSpaces contexte de chiffrement
- WorkSpaces Autorisation d'utiliser une clé KMS en votre nom

Vue d'ensemble du WorkSpaces chiffrement à l'aide de AWS KMS

Lorsque vous créez WorkSpaces avec des volumes chiffrés, WorkSpaces utilisez Amazon Elastic Block Store (Amazon EBS) pour créer et gérer ces volumes. Les deux services utilisent votre AWS KMS key pour travailler avec les volumes chiffrés. Pour plus d'information sur le chiffrement des volumes EBS, consultez la documentation suivante :

- Comment Amazon Elastic Block Store (Amazon EBS) utilise AWS KMS dans ce guide
- Chiffrement Amazon EBS dans le Guide de l'utilisateur Amazon EC2 pour les instances Windows

Lorsque vous lancez WorkSpaces avec des volumes chiffrés, le end-to-end processus fonctionne comme suit :

1. Vous spécifiez la clé KMS à utiliser pour le chiffrement ainsi que WorkSpace l'utilisateur et le répertoire. Cette action crée une autorisation qui permet d'utiliser votre clé KMS uniquement WorkSpaces à cette fin, c' WorkSpaceest-à-dire uniquement pour l'utilisateur et le répertoire WorkSpace associés à l'utilisateur et au répertoire spécifiés.
2. WorkSpaces crée un volume EBS chiffré pour le WorkSpace et spécifie la clé KMS à utiliser ainsi que l'utilisateur et le répertoire du volume (les mêmes informations que celles que vous avez spécifiées àStep 1). Cette action crée une autorisation qui permet à Amazon EBS d'utiliser votre clé KMS uniquement pour ce volume, c'est-à-dire uniquement pour l'utilisateur WorkSpace et le répertoire WorkSpace associés à l'utilisateur et au répertoire spécifiés, et uniquement pour le volume spécifié.
3. Amazon EBS demande une clé de données de volume chiffrée sous votre clé KMS et spécifie l'ID de WorkSpace l'utilisateur `Sid` et du répertoire ainsi que l'ID du volume comme contexte de chiffrement.
4. AWS KMS crée une nouvelle clé de données, la chiffre sous votre clé KMS, puis envoie la clé de données chiffrée à Amazon EBS.
5. WorkSpaces utilise Amazon EBS pour associer le volume chiffré à votre WorkSpace. Amazon EBS envoie la clé de données chiffrée AWS KMS à une `Decrypt`demande et spécifie celle de l' WorkSpace utilisateur`Sid`, son ID de répertoire et l'ID de volume, qui est utilisé comme contexte de chiffrement.
6. AWS KMS utilise votre clé KMS pour déchiffrer la clé de données, puis envoie la clé de données en texte brut à Amazon EBS.
7. Amazon EBS se sert de la clé de données en texte brut pour chiffrer toutes les données en direction et en provenance du volume chiffré. Amazon EBS conserve la clé de données en texte brut en mémoire tant que le volume est attaché au. WorkSpace
8. Amazon EBS stocke la clé de données cryptée (reçue àStep 4) avec les métadonnées du volume pour une utilisation future au cas où vous redémarreriez ou reconstruisez le WorkSpace.
9. Lorsque vous utilisez le AWS Management Console pour supprimer un WorkSpace (ou que vous utilisez l'`TerminateWorkspaces`action dans l' WorkSpaces API) WorkSpaces et qu'Amazon EBS retire les autorisations qui lui permettaient d'utiliser votre clé KMS à cette WorkSpace fin.

WorkSpaces contexte de chiffrement

WorkSpaces ne vous utilise pas AWS KMS key directement pour des opérations cryptographiques (telles que `Encrypt`,, `DecryptGenerateDataKey`, etc.), ce qui signifie AWS KMS qu' WorkSpaces

il n'envoie pas de demandes incluant un contexte de chiffrement. Toutefois, lorsqu'Amazon EBS demande une clé de données chiffrée pour les volumes chiffrés de votre WorkSpaces (Step 3dans leVue d'ensemble du WorkSpaces chiffrement à l'aide de AWS KMS) et lorsqu'il demande une copie en texte clair de cette clé de données (Step 5), il inclut le contexte de chiffrement dans la demande. Le contexte de chiffrement fournit des données authentifiées supplémentaires (AAD) qu'AWS KMS utilise pour garantir l'intégrité des données. Le contexte de chiffrement est également écrit dans les fichiers journaux AWS CloudTrail qui vous aident à comprendre pourquoi une AWS KMS key donnée a été utilisée. Amazon EBS utilise les éléments suivants comme contexte de chiffrement :

- Le nom `sid` de AWS Directory Service l'utilisateur associé au WorkSpace
- L'ID du AWS Directory Service répertoire associé au WorkSpace
- L'ID de volume du volume chiffré

L'exemple suivant montre une représentation JSON du contexte de chiffrement qu'Amazon EBS utilise :

```
{
  "aws:workspaces:sid-directoryid":
 "[S-1-5-21-277731876-1789304096-451871588-1107]@[d-1234abcd01]",
  "aws:ebs:id": "vol-1234abcd"
}
```

WorkSpaces Autorisation d'utiliser une clé KMS en votre nom

Vous pouvez protéger les données de votre espace de travail en utilisant le Clé gérée par AWS for WorkSpaces (aws/workspaces) ou une clé gérée par le client. Si vous utilisez une clé gérée par le client, vous devez WorkSpaces autoriser l'utilisation de la clé KMS au nom des WorkSpaces administrateurs de votre compte. Le Clé gérée par AWS formulaire WorkSpaces dispose des autorisations requises par défaut.

Pour préparer votre clé gérée par le client à utiliser avec WorkSpaces, suivez la procédure suivante.

1. Ajoutez les WorkSpaces administrateurs à la liste des utilisateurs clés dans la politique clé de la clé KMS
2. Donnez aux WorkSpaces administrateurs des autorisations supplémentaires grâce à une politique IAM

WorkSpaces les administrateurs ont également besoin d'une autorisation d'utilisation WorkSpaces. Pour plus d'informations sur ces autorisations, consultez la section Contrôle de l'accès aux WorkSpaces ressources dans le guide d' WorkSpaces administration Amazon.

Partie 1 : Ajouter WorkSpaces des administrateurs aux utilisateurs principaux d'une clé KMS

Pour donner WorkSpaces aux administrateurs les autorisations dont ils ont besoin, vous pouvez utiliser l'API AWS Management Console ou l'AWS KMSAPI.

Pour ajouter des WorkSpaces administrateurs en tant qu'utilisateurs clés pour une clé KMS (console)

1. Connectez-vous à AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer le paramètre Région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Dans le volet de navigation, choisissez Clés gérées par le client.
4. Choisissez l'ID de clé ou l'alias de votre clé gérée par le client préférée.
5. Choisissez l'onglet Stratégie de clé. Sous Utilisateurs de clé, choisissez Ajouter.
6. Dans la liste des utilisateurs et des rôles IAM, sélectionnez les utilisateurs et les rôles correspondant à vos WorkSpaces administrateurs, puis choisissez Joindre.

Pour ajouter des WorkSpaces administrateurs en tant qu'utilisateurs clés pour une clé KMS (AWS KMSAPI)

1. Utilisez cette GetKeyPolicyopération pour obtenir la politique clé existante, puis enregistrez le document de stratégie dans un fichier.
2. Ouvrez le document de stratégie dans votre éditeur de texte préféré. Ajoutez les utilisateurs et les rôles IAM correspondant à vos WorkSpaces administrateurs aux déclarations de politique qui accordent des autorisations aux utilisateurs clés. Ensuite, enregistrez le fichier.
3. Utilisez l'PutKeyPolicyopération pour appliquer la politique de clé à la clé KMS.

Partie 2 : Accorder des autorisations supplémentaires WorkSpaces aux administrateurs

Si vous utilisez une clé gérée par le client pour protéger vos WorkSpaces données, outre les autorisations définies dans la section des utilisateurs clés de la politique de clé par défaut, WorkSpaces les administrateurs doivent être autorisés à créer des autorisations sur la clé KMS. De plus, s'ils utilisent le AWS Management Consolepour créer WorkSpaces avec des volumes chiffrés, WorkSpaces les administrateurs doivent être autorisés à répertorier les alias et les clés de liste. Pour de plus amples informations sur la création et la modification de politiques d'utilisateurs IAM, veuillez consulter Politiques gérées et politiques en ligne dans le Guide de l'utilisateur IAM.

Pour accorder ces autorisations à vos WorkSpaces administrateurs, utilisez une politique IAM. Ajoutez une déclaration de politique similaire à l'exemple suivant à la stratégie IAM de chaque WorkSpaces administrateur. Remplacez l'exemple d'ARN de clé KMS (*arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab*) par un ARN valide. Si vos WorkSpaces administrateurs utilisent uniquement l' WorkSpaces API (et non la console), vous pouvez omettre la deuxième déclaration de politique avec les `"kms:ListKeys"` autorisations `"kms:ListAliases"` et.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:ListAliases",
        "kms:ListKeys"
      ],
      "Resource": "*"
    }
  ]
}
```

Programmation de l'API AWS KMS

Vous pouvez utiliser l'API AWS KMS pour créer et gérer des clés KMS et des fonctionnalités spéciales, telles que des magasins de clés personnalisés, et utiliser des clés KMS dans les opérations de chiffrement. Pour plus d'informations, consultez la référence d'API AWS Key Management Service.

L'exemple de code utilisé dans les rubriques suivantes illustre comment utiliser les kits SDK AWS pour appeler l'API AWS KMS.

Pour de plus amples informations sur l'utilisation de la console AWS KMS pour effectuer certaines de ces tâches, veuillez consulter Gestion de clés .

Rubriques

- Création d'un client
- Utilisation de clés
- Utilisation des alias
- Chiffrement et déchiffrement des clés de données
- Utilisation de politiques de clé
- Utilisation d'octrois
- Tester vos appels d'API AWS KMS
- cohérence à terme AWS KMS

Création d'un client

Pour utiliser le AWS SDK for Java, le AWS SDK for .NET, le AWS SDK for Python (Boto3)AWS SDK for Ruby, le ou le AWS SDK for PHPAWSSDK JavaScript dans Node.js afin d'écrire du code utilisant l'API AWS Key Management Service (AWS KMS), commencez par créer un AWS KMS client.

L'objet client que vous créez est utilisé dans l'exemple de code figurant dans les rubriques suivantes.

Java

Pour créer un client AWS KMS dans Java, utilisez le générateur client.

```
AWSKMS kmsClient = AWSKMSClientBuilder.standard().build();
```

Pour plus d'informations sur l'utilisation du générateur client Java, consultez les ressources suivantes.

- Fluent Client Builders dans le Blog des développeurs AWS
- Création de clients de service dans le Manuel du développeur AWS SDK for Java
- AWSKMSClientBuilder dans la Référence d'API AWS SDK for Java

C#

```
AmazonKeyManagementServiceClient kmsClient = new AmazonKeyManagementServiceClient();
```

Python

```
kms_client = boto3.client('kms')
```

Ruby

```
require 'aws-sdk-kms'  # in v2: require 'aws-sdk'

kmsClient = Aws::KMS::Client.new
```

PHP

Pour créer un client AWS KMS dans PHP, utilisez un objet client AWS KMS, et spécifiez la version `2014-11-01`. Pour plus d'informations, veuillez consulter la classe KMSClient dans la référence d'API AWS SDK for PHP.

```
// Create a KMSClient
$KmsClient = new Aws\Kms\KmsClient([
    'profile' => 'default',
    'version' => '2014-11-01',
    'region'  => 'us-east-1'
]);
```

Node.js

```
const kmsClient = new AWS.KMS();
```

Utilisation de clés

Les exemples de cette rubrique utilisent l'API AWS KMS pour créer, afficher, activer et désactiver des AWS KMS keys AWS KMS, et pour générer des clés de données.

Rubriques

- Création d'une clé KMS
- Génération d'une clé de données
- Affichage d'un AWS KMS key
- Obtention des ID de clé et des ARN de clé des clés KMS
- Activer AWS KMS keys
- Désactivation de AWS KMS key

Création d'une clé KMS

Pour créer une AWS KMS key(clé KMS), utilisez l'CreateKeyopération. Les exemples de cette section créent une clé KMS de chiffrement symétrique. Le paramètre `Description` utilisé dans ces exemples est facultatif.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Pour obtenir de l'aide sur la création des clés KMS dans la console AWS KMS, veuillez consulter Création de clés.

Java

Pour obtenir des détails, veuillez consulter la méthode createKey dans la Référence d'API AWS SDK for Java.

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest().withDescription(desc);
CreateKeyResult result = kmsClient.createKey(req);
```

C#

Pour obtenir des détails, consultez la méthode CreateKey dans AWS SDK for .NET.

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest()
{
    Description = desc
};
CreateKeyResponse response = kmsClient.CreateKey(req);
```

Python

Pour obtenir des détails, consultez la méthode `create_key` dans AWS SDK for Python (Boto3).

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kms_client.create_key(
    Description=desc
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `create_key` dans le kit AWS SDK for Ruby.

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kmsClient.create_key({
  description: desc
})
```

PHP

Pour obtenir des détails, consultez la méthode CreateKey dans AWS SDK for PHP.

```
// Create a KMS key
//
$desc = "Key for protecting critical data";

$result = $KmsClient->createKey([
    'Description' => $desc
]);
```

Node.js

Pour plus de détails, consultez la propriété CreateKey dans le AWSSDK pour JavaScript Node.js.

```
// Create a KMS key
//
const Description = 'Key for protecting critical data';

kmsClient.createKey({ Description }, (err, data) => {
  ...
});
```

PowerShell

Pour créer une clé KMS dans PowerShell, utilisez l'KmsKeyapplet de commande New-.

```
# Create a KMS key

$desc = 'Key for protecting critical data'
New-KmsKey -Description $desc
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Génération d'une clé de données

Pour générer une clé de données symétrique, utilisez l'GenerateDataKeyopération. Cette opération renvoie une clé de données en texte brut et une copie de cette clé de données chiffrée sous un clé KMS de chiffrement symétrique que vous spécifiez. Vous devez spécifier `KeySpec` ou `NumberOfBytes` (mais pas les deux) dans chaque commande.

Pour obtenir de l'aide sur l'utilisation de la clé de données pour chiffrer les données, consultez le AWS Encryption SDK. Vous pouvez également utiliser la clé de données dans les opérations HMAC.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour plus de détails, consultez la generateDataKey méthode dans la référence de AWS SDK for Java l'API.

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

GenerateDataKeyRequest dataKeyRequest = new GenerateDataKeyRequest();
dataKeyRequest.setKeyId(keyId);
dataKeyRequest.setKeySpec("AES_256");

GenerateDataKeyResult dataKeyResult = kmsClient.generateDataKey(dataKeyRequest);

ByteBuffer plaintextKey = dataKeyResult.getPlaintext();

ByteBuffer encryptedKey = dataKeyResult.getCiphertextBlob();
```

C#

Pour obtenir des détails, consultez la méthode GenerateDataKey dans AWS SDK for .NET.

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
GenerateDataKeyRequest dataKeyRequest = new GenerateDataKeyRequest()
{
    KeyId = keyId,
    KeySpec = DataKeySpec.AES_256
};
```

```
GenerateDataKeyResponse dataKeyResponse = kmsClient.GenerateDataKey(dataKeyRequest);

MemoryStream plaintextKey = dataKeyResponse.Plaintext;

MemoryStream encryptedKey = dataKeyResponse.CiphertextBlob;
```

Python

Pour obtenir des détails, consultez la méthode `generate_data_key` dans AWS SDK for Python (Boto3).

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.generate_data_key(
    KeyId=key_id,
    KeySpec='AES_256'
)

plaintext_key = response['Plaintext']

encrypted_key = response['CiphertextBlob']
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `generate_data_key` dans le kit AWS SDK for Ruby.

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.generate_data_key({
  key_id: key_id,
  key_spec: 'AES_256'
})
```

```
plaintext_key = response.plaintext

encrypted_key = response.ciphertext_blob
```

PHP

Pour obtenir des détails, consultez la méthode GenerateDataKey dans AWS SDK for PHP.

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$keySpec = 'AES_256';

$result = $KmsClient->generateDataKey([
    'KeyId' => $keyId,
    'KeySpec' => $keySpec,
]);

$plaintextKey = $result['Plaintext'];

$encryptedKey = $result['CiphertextBlob'];
```

Node.js

Pour plus de détails, consultez la generateDataKey propriété dans le AWSSDK pour JavaScript dans Node.js.

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const KeySpec = 'AES_256';
kmsClient.generateDataKey({ KeyId, KeySpec }, (err, data) => {
  if (err) console.log(err, err.stack);
  else {
    const { CiphertextBlob, Plaintext } = data;
      ...
  }
});
```

PowerShell

Pour générer une clé de données symétrique, utilisez l'applet de commande New-KMS DataKey.

Dans la sortie, la clé en texte brut (dans la `Plaintext` propriété) et la clé cryptée (dans la `CiphertextBlob` propriété) sont MemoryStreamdes objets. Pour les convertir en chaînes, utilisez les méthodes de la `MemoryStream` classe, une applet de commande ou une fonction qui convertit les `MemoryStream` objets en chaînes, telles que les fonctions ConvertFrom-MemoryStream et ConvertFrom-Base64 du module Convert.

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$keySpec = 'AES_256'

$response = New-KmsDataKey -KeyId $keyId -KeySpec $keySpec
$plaintextKey = $response.Plaintext
$encryptedKey = $response.CiphertextBlob
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Affichage d'un AWS KMS key

Pour obtenir des informations détaillées sur unAWS KMS key, y compris l'ARN et l'état de la clé KMS, utilisez l'DescribeKeyopération.

`DescribeKey` n'obtient pas d'alias. Pour obtenir des alias, utilisez l'ListAliasesopération. Pour obtenir des exemples, consultez Utilisation des alias.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Pour obtenir de l'aide sur l'affichage des clés KMS dans la console AWS KMS, veuillez consulter Affichage des clés.

Java

Pour obtenir des détails, veuillez consulter la méthode describeKey dans la Référence d'API AWS SDK for Java.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DescribeKeyRequest req = new DescribeKeyRequest().withKeyId(keyId);
DescribeKeyResult result = kmsClient.describeKey(req);
```

C#

Pour obtenir des détails, consultez la méthode DescribeKey dans AWS SDK for .NET.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DescribeKeyRequest describeKeyRequest = new DescribeKeyRequest()
{
    KeyId = keyId
};

DescribeKeyResponse describeKeyResponse = kmsClient.DescribeKey(describeKeyRequest);
```

Python

Pour obtenir des détails, consultez la méthode `describe_key` dans AWS SDK for Python (Boto3).

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.describe_key(
```

```
    KeyId=key_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `describe_key` dans le kit AWS SDK for Ruby.

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.describe_key({
  key_id: key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode DescribeKey dans AWS SDK for PHP.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->describeKey([
    'KeyId' => $keyId,
]);
```

Node.js

Pour plus de détails, consultez la propriété DescribeKey dans AWSle SDK JavaScript pour Node.js.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.describeKey({ KeyId }, (err, data) => {
```

```
  ...
});
```

PowerShell

Pour obtenir des informations détaillées sur une clé KMS, utilisez l'KmsKeyapplet de commande Get-.

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Get-KmsKey -KeyId $keyId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Obtention des ID de clé et des ARN de clé des clés KMS

Pour obtenir les identifiants clés et les ARN clés duAWS KMS keys, utilisez l'ListKeysopération. Ces exemples utilisent le paramètre facultatif `Limit`, qui définit le nombre maximal de clés KMS renvoyées dans chaque appel. Pour obtenir de l'aide sur l'identification d'une clé KMS dans des opérations AWS KMS, veuillez consulter Identifiants clés () KeyId.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Pour obtenir de l'aide sur la recherche d'ID de clé et d'ARN de clé dans la console AWS KMS, veuillez consulter Recherche de l'ID et de l'ARN d'une clé.

Java

Pour obtenir des détails, veuillez consulter la méthode listKeys dans la Référence d'API AWS SDK for Java.

```
// List KMS keys in this account
//
Integer limit = 10;
```

```
ListKeysRequest req = new ListKeysRequest().withLimit(limit);
ListKeysResult result = kmsClient.listKeys(req);
```

C#

Pour obtenir des détails, consultez la méthode ListKeys dans AWS SDK for .NET.

```
// List KMS keys in this account
//
int limit = 10;

ListKeysRequest listKeysRequest = new ListKeysRequest()
{
    Limit = limit
};
ListKeysResponse listKeysResponse = kmsClient.ListKeys(listKeysRequest);
```

Python

Pour obtenir des détails, consultez la méthode `list_keys` dans AWS SDK for Python (Boto3).

```
# List KMS keys in this account

response = kms_client.list_keys(
    Limit=10
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `list_keys` dans le kit AWS SDK for Ruby.

```
# List KMS keys in this account

response = kmsClient.list_keys({
  limit: 10
})
```

PHP

Pour obtenir des détails, consultez la méthode ListKeys dans AWS SDK for PHP.

```
// List KMS keys in this account
```

```
//
$limit = 10;

$result = $KmsClient->listKeys([
    'Limit' => $limit,
]);
```

Node.js

Pour plus de détails, consultez la propriété ListKeys dans le AWSSDK pour JavaScript Node.js.

```
// List KMS keys in this account
//
const Limit = 10;
kmsClient.listKeys({ Limit }, (err, data) => {
  ...
});
```

PowerShell

Pour obtenir l'ID de clé et l'ARN de toutes les clés KMS du compte et de la région, utilisez l'KmsKeyListapplet de commande Get-.

Pour limiter le nombre d'objets en sortie, cet exemple utilise l'applet de commande Select-Object au lieu du paramètre `Limit`, obsolète dans cette applet de commande. Pour obtenir de l'aide sur la pagination de sortie dans AWS Tools for PowerShell, veuillez consulter le billet de blog Output Pagination with AWS Tools for PowerShell.

```
# List KMS keys in this account

$limit = 10
Get-KmsKeyList | Select-Object -First $limit
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Activer AWS KMS keys

Pour activer une personne désactivéeAWS KMS key, utilisez l'EnableKeyopération.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Pour obtenir de l'aide sur l'activation et la désactivation des clés KMS dans la console AWS KMS, veuillez consulter Activation et désactivation des clés.

Java

Pour plus de détails sur l'implémentation Java, veuillez consulter la méthode enableKey dans la Référence d'API AWS SDK for Java.

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

EnableKeyRequest req = new EnableKeyRequest().withKeyId(keyId);
kmsClient.enableKey(req);
```

C#

Pour obtenir des détails, consultez la méthode EnableKey dans AWS SDK for .NET.

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

EnableKeyRequest enableKeyRequest = new EnableKeyRequest()
{
    KeyId = keyId
};
kmsClient.EnableKey(enableKeyRequest);
```

Python

Pour obtenir des détails, consultez la méthode `enable_key` dans AWS SDK for Python (Boto3).

```
# Enable a KMS key
```

```
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.enable_key(
    KeyId=key_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `enable_key` dans le kit AWS SDK for Ruby.

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.enable_key({
  key_id: key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode EnableKey dans AWS SDK for PHP.

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->enableKey([
    'KeyId' => $keyId,
]);
```

Node.js

Pour plus de détails, consultez la propriété EnableKey dans AWSle SDK JavaScript pour Node.js.

```
// Enable a KMS key
//
```

```
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.enableKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour activer une clé KMS, utilisez l'KmsKeyapplet de commande Enable-.

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Enable-KmsKey -KeyId $keyId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Désactivation de AWS KMS key

Pour désactiver une clé KMS, utilisez l'DisableKeyopération. La désactivation d'une clé KMS empêche son utilisation dans des opérations de chiffrement.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Pour obtenir de l'aide sur l'activation et la désactivation des clés KMS dans la console AWS KMS, veuillez consulter Activation et désactivation des clés.

Java

Pour obtenir des détails, veuillez consulter la méthode disableKey dans la Référence d'API AWS SDK for Java.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
```

```
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DisableKeyRequest req = new DisableKeyRequest().withKeyId(keyId);
kmsClient.disableKey(req);
```

C#

Pour obtenir des détails, consultez la méthode DisableKey dans AWS SDK for .NET.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DisableKeyRequest disableKeyRequest = new DisableKeyRequest()
{
    KeyId = keyId
};
kmsClient.DisableKey(disableKeyRequest);
```

Python

Pour obtenir des détails, consultez la méthode `disable_key` dans AWS SDK for Python (Boto3).

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.disable_key(
    KeyId=key_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `disable_key` dans le kit AWS SDK for Ruby.

```
# Disable a KMS key
```

```
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.disable_key({
  key_id: key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode DisableKey dans AWS SDK for PHP.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->disableKey([
    'KeyId' => $keyId,
]);
```

Node.js

Pour plus de détails, consultez la propriété DisableKey dans AWSle SDK JavaScript pour Node.js.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.disableKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour désactiver une clé KMS, utilisez l'KmsKeyapplet de commande Disable-.

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
```

```
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Disable-KmsKey -KeyId $keyId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Utilisation des alias

Les exemples de cette rubrique utilisent l'API AWS KMS pour créer, afficher, mettre à jour et supprimer des alias. Pour en savoir plus sur les alias, veuillez consulter the section called “Utilisation des alias”.

Rubriques

- Création d'un alias
- Établissement de la liste des alias
- Mise à jour d'un alias
- Suppression d'un alias

Création d'un alias

Lorsque vous créez une AWS KMS key dans la AWS Management Console, vous devez créer un alias pour cela. Toutefois, l'CreateKeyopération qui crée une clé KMS ne crée pas d'alias.

Pour créer un alias, utilisez l'CreateAliasopération. L'alias doit être unique dans le compte et la région. Vous ne pouvez pas créer un alias commençant par `aws/`. Le préfixe `aws/` est réservé par Amazon Web Services pour Clés gérées par AWS.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode createAlias dans la Référence d'API AWS SDK for Java.

```
// Create an alias for a KMS key
```

```
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CreateAliasRequest req = new
 CreateAliasRequest().withAliasName(aliasName).withTargetKeyId(targetKeyId);
kmsClient.createAlias(req);
```

C#

Pour obtenir des détails, consultez la méthode CreateAlias dans AWS SDK for .NET.

```
// Create an alias for a KMS key
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CreateAliasRequest createAliasRequest = new CreateAliasRequest()
{
    AliasName = aliasName,
    TargetKeyId = targetKeyId
};
kmsClient.CreateAlias(createAliasRequest);
```

Python

Pour obtenir des détails, consultez la méthode `create_alias` dans AWS SDK for Python (Boto3).

```
# Create an alias for a KMS key

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
target_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.create_alias(
    AliasName=alias_name,
    TargetKeyId=key_id
```

```
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `create_alias` dans le kit AWS SDK for Ruby.

```
# Create an alias for a KMS key

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
target_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.create_alias({
  alias_name: alias_name,
  target_key_id: target_key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode CreateAlias dans AWS SDK for PHP.

```
// Create an alias for a KMS key
//
$aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';


$result = $KmsClient->createAlias([
    'AliasName' => $aliasName,
    'TargetKeyId' =>  $keyId,
]);
```

Node.js

Pour plus de détails, consultez la propriété CreateAlias dans le AWSSDK pour JavaScript Node.js.

```
// Create an alias for a KMS key
```

```
//
const AliasName = 'alias/projectKey1';

// Replace the following example key ARN with a valid key ID or key ARN
const TargetKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.createAlias({ AliasName, TargetKeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour créer un alias, utilisez l'applet de commande New-KMSAlias. Le nom de l'alias est sensible à la casse.

```
# Create an alias for a KMS key

$aliasName = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
$targetKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

New-KMSAlias -TargetKeyId $targetKeyId -AliasName $aliasName
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Établissement de la liste des alias

Pour répertorier les alias du compte et de la région, utilisez l'ListAliasesopération.

Par défaut, la commande ListAliases retourne tous les alias figurant dans le compte et la région. Cela inclut les alias que vous avez créés et associés à vos clés KMS gérées par le client et les alias créés par AWS et associés à vos Clés gérées par AWS. La réponse peut également inclure des alias ne disposant pas de champ `TargetKeyId`. Ces alias prédéfinis ont été créés par AWS qui ne les a pas encore associés à une clé KMS.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour plus de détails sur l'implémentation Java, veuillez consulter la méthode listAliases dans la Référence d'API AWS SDK for Java.

```
// List the aliases in this Compte AWS
//
Integer limit = 10;

ListAliasesRequest req = new ListAliasesRequest().withLimit(limit);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

Pour obtenir des détails, consultez la méthode ListAliases dans AWS SDK for .NET.

```
// List the aliases in this Compte AWS
//
int limit = 10;

ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    Limit = limit
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

Pour obtenir des détails, consultez la méthode `list_aliases` dans AWS SDK for Python (Boto3).

```
# List the aliases in this Compte AWS

response = kms_client.list_aliases(
    Limit=10
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `list_aliases` dans le kit AWS SDK for Ruby.

```
# List the aliases in this Compte AWS

response = kmsClient.list_aliases({
  limit: 10
})
```

PHP

Pour obtenir des détails, consultez la méthode List Aliases dans le kit AWS SDK for PHP.

```
// List the aliases in this Compte AWS
//
$limit = 10;

$result = $KmsClient->listAliases([
    'Limit' => $limit,
]);
```

Node.js

Pour plus de détails, consultez la propriété ListAliases dans AWS le SDK pour Node.js. JavaScript

```
// List the aliases in this Compte AWS
//
const Limit = 10;
kmsClient.listAliases({ Limit }, (err, data) => {
  ...
});
```

PowerShell

Pour répertorier les alias du compte et de la région, utilisez l'applet de commande AliasListGet-KMS.

Pour limiter le nombre d'objets en sortie, cet exemple utilise l'applet de commande Select-Object au lieu du paramètre `Limit`, obsolète dans cette applet de commande. Pour obtenir de l'aide sur la pagination de sortie dans AWS Tools for PowerShell, veuillez consulter le billet de blog Output Pagination with AWS Tools for PowerShell.

```
# List the aliases in this Compte AWS
$limit = 10
```

```
$result = Get-KMSAliasList | Select-Object -First $limit
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Pour répertorier uniquement les alias associés à une clé KMS spécifique, utilisez le paramètre `KeyId`. Sa valeur peut être l'ID de clé ou l'ARN de clé de n'importe quelle clé KMS dans la région. Vous ne pouvez pas spécifier de nom d'alias ou d'ARN d'alias.

Java

Pour plus de détails sur l'implémentation Java, veuillez consulter la méthode listAliases dans la Référence d'API AWS SDK for Java.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListAliasesRequest req = new ListAliasesRequest().withKeyId(keyId);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

Pour obtenir des détails, consultez la méthode ListAliases dans AWS SDK for .NET.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    KeyId = keyId
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

Pour obtenir des détails, consultez la méthode `list_aliases` dans AWS SDK for Python (Boto3).

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_aliases(
    KeyId=key_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `list_aliases` dans le kit AWS SDK for Ruby.

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_aliases({
  key_id: key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode List Aliases dans le kit AWS SDK for PHP.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->listAliases([
    'KeyId' => $keyId,
```

```
]);
```

Node.js

Pour plus de détails, consultez la propriété ListAliases dans AWS le SDK pour Node.js. JavaScript

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.listAliases({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour répertorier les alias d'une clé KMS, utilisez le `KeyId` paramètre de l'applet de commande AliasListGet-KMS.

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

$response = Get-KmsAliasList -KeyId $keyId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools. KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Mise à jour d'un alias

Pour associer un alias existant à une autre clé KMS, utilisez l'UpdateAliasopération.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour plus de détails sur l'implémentation Java, veuillez consulter la méthode updateAlias dans la Référence d'API AWS SDK for Java.

```
// Updating an alias
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

UpdateAliasRequest req = new UpdateAliasRequest()
     .withAliasName(aliasName)
     .withTargetKeyId(targetKeyId);

kmsClient.updateAlias(req);
```

C#

Pour obtenir des détails, consultez la méthode UpdateAlias dans AWS SDK for .NET.

```
// Updating an alias
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

UpdateAliasRequest updateAliasRequest = new UpdateAliasRequest()
{
    AliasName = aliasName,
    TargetKeyId = targetKeyId
};

kmsClient.UpdateAlias(updateAliasRequest);
```

Python

Pour obtenir des détails, consultez la méthode `update_alias` dans AWS SDK for Python (Boto3).

```
# Updating an alias
```

```
alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

response = kms_client.update_alias(
    AliasName=alias_name,
    TargetKeyID=key_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `update_alias` dans le kit AWS SDK for Ruby.

```
# Updating an alias

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

response = kmsClient.update_alias({
  alias_name: alias_name,
  target_key_id: key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode UpdateAlias dans AWS SDK for PHP.

```
// Updating an alias
//
$aliasName = "alias/projectKey1";

// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';

$result = $KmsClient->updateAlias([
    'AliasName' => $aliasName,
    'TargetKeyId' =>  $keyId,
```

```
]);
```

Node.js

Pour plus de détails, consultez la propriété UpdateAlias dans AWSle SDK JavaScript pour Node.js.

```
// Updating an alias
//
const AliasName = 'alias/projectKey1';

// Replace the following example key ARN with a valid key ID or key ARN
const TargetKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';
kmsClient.updateAlias({ AliasName, TargetKeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour modifier la clé KMS associée à un alias, utilisez l'applet de commande Update-KMSAlias. Le nom de l'alias est sensible à la casse.

L'applet de commande `Update-KMSAlias` ne renvoie aucune sortie. Pour vérifier que la commande a fonctionné, utilisez l'applet de commande Get-KMS AliasList.

```
# Updating an alias

$aliasName = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

Update-KMSAlias -AliasName $aliasName -TargetKeyID $keyId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d’informations, consultez le Guide de l’utilisateur AWS Tools for Windows PowerShell.

Suppression d'un alias

Pour supprimer un alias, utilisez l'DeleteAliasopération. La suppression d'un alias n'a aucun effet sur la clé KMS associée.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode deleteAlias dans la Référence d'API AWS SDK for Java.

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest req  = new DeleteAliasRequest().withAliasName(aliasName);
kmsClient.deleteAlias(req);
```

C#

Pour obtenir des détails, consultez la méthode DeleteAlias dans AWS SDK for .NET.

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest deleteAliasRequest = new DeleteAliasRequest()
{
    AliasName = aliasName
};
kmsClient.DeleteAlias(deleteAliasRequest);
```

Python

Pour obtenir des détails, consultez la méthode `delete_alias` dans AWS SDK for Python (Boto3).

```
# Delete an alias for a KMS key
```

```
alias_name = 'alias/projectKey1'

response = kms_client.delete_alias(
    AliasName=alias_name
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `delete_alias` dans le kit AWS SDK for Ruby.

```
# Delete an alias for a KMS key

alias_name = 'alias/projectKey1'

response = kmsClient.delete_alias({
  alias_name: alias_name
})
```

PHP

Pour obtenir des détails, consultez la méthode DeleteAlias dans AWS SDK for PHP.

```
// Delete an alias for a KMS key
//
$aliasName = "alias/projectKey1";

$result = $KmsClient->deleteAlias([
    'AliasName' => $aliasName,
]);
```

Node.js

Pour plus de détails, consultez la propriété DeleteAlias (propriété) dans AWSIe SDK JavaScript pour Node.js.

```
// Delete an alias for a KMS key
//
const AliasName = 'alias/projectKey1';
kmsClient.deleteAlias({ AliasName }, (err, data) => {
  ...
});
```

PowerShell

Pour supprimer un alias, utilisez l'applet de commande Remove-KMSAlias. Le nom de l'alias est sensible à la casse.

Comme cette applet de commande supprime définitivement l'alias, elle vous PowerShell invite à confirmer la commande. Le paramètre `ConfirmImpact` étant défini sur `High`, vous ne pouvez pas utiliser `ConfirmPreference` pour supprimer cette invite. Si vous devez supprimer l'invite de confirmation, ajoutez le paramètre `Confirm` courant avec la valeur `$false`, par exemple : `-Confirm:$false`.

L' applet de commande `Remove-KMSAlias` ne renvoie aucune sortie. Pour vérifier l'efficacité de la commande, utilisez l'applet de commande Get-KMS AliasList.

```
# Delete an alias for a KMS key

$aliasName = 'alias/projectKey1'
Remove-KMSAlias -AliasName $aliasName
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Chiffrement et déchiffrement des clés de données

Les exemples présentés dans cette rubrique utilisent le chiffrement, le déchiffrement et les ReEncryptopérations de l'AWS KMSAPI.

Ces opérations sont conçues pour chiffrer et déchiffrer des clés de données. Elles utilisent une AWS KMS keys dans les opérations de chiffrement et ne peuvent pas accepter plus de 4 Ko (4 096 octets) de données. Même si vous pouvez les utiliser pour chiffrer des petits volumes de données, comme un mot de passe ou une clé RSA, ces opérations ne sont pas conçues pour chiffrer des données d'application.

Pour chiffrer des données d'application, utilisez les fonctions de chiffrement côté serveur d'un service AWS, une bibliothèque de chiffrement côté client comme AWS Encryption SDK ou le client de chiffrement Amazon S3.

Rubriques

- Chiffrement d'une clé de données
- Déchiffrement d'une clé de données
- Rechiffrement d'une clé de données sous une autre AWS KMS key

Chiffrement d'une clé de données

L'opération Encrypt est conçue pour chiffrer des clés de données, mais elle n'est pas fréquemment utilisée. Les GenerateDataKeyWithoutPlaintextopérations GenerateDataKeyet renvoient des clés de données chiffrées. Vous pouvez utiliser cette méthode lorsque vous déplacez des données chiffrées vers une région différente et que vous souhaitez chiffrer leur clé de données à l'aide d'une clé KMS dans la nouvelle région.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode encrypt dans la Référence d'API AWS SDK for Java.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
ByteBuffer plaintext = ByteBuffer.wrap(new byte[]{1,2,3,4,5,6,7,8,9,0});

EncryptRequest req = new EncryptRequest().withKeyId(keyId).withPlaintext(plaintext);
ByteBuffer ciphertext = kmsClient.encrypt(req).getCiphertextBlob();
```

C#

Pour obtenir des détails, consultez la méthode Encrypt dans le kit AWS SDK for .NET.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
MemoryStream plaintext = new MemoryStream();
plaintext.Write(new byte[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }, 0, 10);
```

```
EncryptRequest encryptRequest = new EncryptRequest()
{
    KeyId = keyId,
    Plaintext = plaintext
};
MemoryStream ciphertext = kmsClient.Encrypt(encryptRequest).CiphertextBlob;
```

Python

Pour obtenir des détails, consultez la méthode encrypt dans la AWS SDK for Python (Boto3).

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
plaintext = b'\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00'

response = kms_client.encrypt(
    KeyId=key_id,
    Plaintext=plaintext
)

ciphertext = response['CiphertextBlob']
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance encrypt dans le kit AWS SDK for Ruby.

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
plaintext = "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00"

response = kmsClient.encrypt({
  key_id: key_id,
  plaintext: plaintext
})

ciphertext = response.ciphertext_blob
```

PHP

Pour obtenir des détails, consultez la méthode Encrypt dans le kit AWS SDK for PHP.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$message = pack('c*',1,2,3,4,5,6,7,8,9,0);

$result = $KmsClient->encrypt([
    'KeyId' => $keyId,
    'Plaintext' => $message,
]);

$ciphertext = $result['CiphertextBlob'];
```

Node.js

Pour plus de détails, consultez la propriété encrypt dans le AWS SDK pour Node.js JavaScript .

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Plaintext = Buffer.from([1, 2, 3, 4, 5, 6, 7, 8, 9, 0]);
kmsClient.encrypt({ KeyId, Plaintext }, (err, data) => {
  if (err) console.log(err, err.stack); // an error occurred
  else {
    const { CiphertextBlob } = data;
      ...
  }
});
```

PowerShell

Pour chiffrer une clé de données sous une clé KMS, utilisez l'applet de commande Invoke-KMSEncrypt. Il renvoie le texte chiffré sous la forme d'un `MemoryStream` (System.IO.MemoryStream) objet. Vous pouvez utiliser l'objet `MemoryStream` comme entrée de l'applet de commande Invoke-KMSDecrypt.

AWS KMS renvoie également des clés de données en tant qu'objets `MemoryStream`. Dans cet exemple, pour simuler une clé de données en texte brut, nous créons un tableau d'octets et l'écrivons dans un objet `MemoryStream`.

Notez que le paramètre `Plaintext` de `Invoke-KMSEncrypt` prend un tableau d'octets (`byte[]`) ; il ne nécessite pas d'objet `MemoryStream`. À partir de AWSPowerShell la version 4.0, les paramètres de tous les AWSPowerShell modules qui acceptent des tableaux d'octets et `MemoryStream` des objets acceptent les tableaux d'octets, les `MemoryStream` objets, les chaînes, les tableaux de chaînes et `FileInfo` (System.IO). FileInfo) objets. Vous pouvez passer n'importe lequel de ces types à `Invoke-KMSEncrypt`.

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Simulate a data key
 # Create a byte array
[byte[]] $bytes = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0

 # Create a MemoryStream
$plaintext = [System.IO.MemoryStream]::new()

 # Add the byte array to the MemoryStream
$plaintext.Write($bytes, 0, $bytes.length)

# Encrypt the simulated data key
$response = Invoke-KMSEncrypt -KeyId $keyId -Plaintext $plaintext

# Get the ciphertext from the response
$ciphertext = $response.CiphertextBlob
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Déchiffrement d'une clé de données

Pour déchiffrer une clé de données, utilisez l'opération Decrypt.

La valeur `ciphertextBlob` que vous spécifiez doit être la valeur du `CiphertextBlob` champ provenant d'une réponse GenerateDataKeyGenerateDataKeyWithoutPlaintext, ou Chiffrer, ou le `PrivateKeyCiphertextBlob` champ d'une GenerateDataKeyPairWithoutPlaintextréponse GenerateDataKeyPairou. Vous pouvez également utiliser l'opération `Decrypt` pour déchiffrer des données chiffrées en dehors de AWS KMS par la clé publique dans une clé KMS asymétrique.

Le paramètre `KeyId` n'est pas requis lors du déchiffrement avec des clés KMS de chiffrement symétriques. AWS KMS peut obtenir la clé KMS utilisée pour chiffrer les données à partir des métadonnées dans le blob de texte chiffré. Toutefois, la spécification de la clé KMS que vous utilisez est une bonne pratique. Cette pratique garantit que vous utilisez la clé KMS prévue et vous empêche de déchiffrer par inadvertance un texte chiffré à l'aide d'une clé KMS non fiable.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode decrypt dans la Référence d'API AWS SDK for Java.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ByteBuffer ciphertextBlob = Place your ciphertext here;

DecryptRequest req = new
 DecryptRequest().withCiphertextBlob(ciphertextBlob).withKeyId(keyId);
ByteBuffer plainText = kmsClient.decrypt(req).getPlaintext();
```

C#

Pour obtenir des détails, consultez la méthode Decrypt dans le kit AWS SDK for .NET.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
```

```
MemoryStream ciphertextBlob = new MemoryStream();
// Write ciphertext to memory stream

DecryptRequest decryptRequest = new DecryptRequest()
{
    CiphertextBlob = ciphertextBlob,
    KeyId = keyId
};
MemoryStream plainText = kmsClient.Decrypt(decryptRequest).Plaintext;
```

Python

Pour obtenir des détails, consultez la méthode decrypt dans la AWS SDK for Python (Boto3).

```
# Decrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
ciphertext = 'Place your ciphertext here'

response = kms_client.decrypt(
    CiphertextBlob=ciphertext,
    KeyId=key_id
)

plaintext = response['Plaintext']
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance decrypt dans le kit AWS SDK for Ruby.

```
# Decrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

ciphertext = 'Place your ciphertext here'
ciphertext_packed = [ciphertext].pack("H*")

response = kmsClient.decrypt({
```

```
  ciphertext_blob: ciphertext_packed,
  key_id: key_id
})

plaintext = response.plaintext
```

PHP

Pour obtenir des détails, consultez la méthode Decrypt dans le kit AWS SDK for PHP.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$ciphertext = 'Place your cipher text blob here';

$result = $KmsClient->decrypt([
     'CiphertextBlob' => $ciphertext,
     'KeyId' => $keyId,
]);

$plaintext = $result['Plaintext'];
```

Node.js

Pour plus de détails, consultez la propriété de déchiffrement dans le AWSSDK pour JavaScript dans Node.js.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const CiphertextBlob = 'Place your cipher text blob here';
kmsClient.decrypt({ CiphertextBlob, KeyId }, (err, data) => {
  if (err) console.log(err, err.stack); // an error occurred
  else {
    const { Plaintext } = data;
    ...
  }
});
```

PowerShell

Pour déchiffrer une clé de données, utilisez l'applet de commande Invoke-KMSEncrypt.

Cette applet de commande renvoie le texte en clair sous la forme d'un `MemoryStream` fichier (System.IO). MemoryStream) objet. Pour le convertir en tableau d'octets, utilisez des applets de commande ou des fonctions qui convertissent des objets `MemoryStream` en tableaux d'octets, telles que les fonctions du module Convert.

Étant donné que cet exemple utilise le texte chiffré renvoyé par une applet de commande de chiffrement AWS KMS, il utilise un objet `MemoryStream` pour la valeur du paramètre `CiphertextBlob`. Cependant, le paramètre `CiphertextBlob` de `Invoke-KMSDecrypt` prend un tableau d'octets (`byte[]`) ; il ne nécessite pas d'objet `MemoryStream`. À partir de AWSPowerShell la version 4.0, les paramètres de tous les AWSPowerShell modules qui acceptent des tableaux d'octets et `MemoryStream` des objets acceptent les tableaux d'octets, les `MemoryStream` objets, les chaînes, les tableaux de chaînes et `FileInfo` (System.IO). FileInfo) objets. Vous pouvez passer n'importe lequel de ces types à `Invoke-KMSDecrypt`.

```
# Decrypt a data key
# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

[System.IO.MemoryStream]$ciphertext = Read-Host 'Place your cipher text blob here'

$response = Invoke-KMSDecrypt -CiphertextBlob $ciphertext -KeyId $keyId
$plaintext = $response.Plaintext
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools. KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Rechiffrement d'une clé de données sous une autre AWS KMS key

Pour déchiffrer une clé de données chiffrée, puis la rechiffrer immédiatement sous une autre cléAWS KMS key, utilisez l'opération. ReEncrypt Les opérations sont effectuées entièrement côté serveur dans AWS KMS, pour que votre texte brut ne soit jamais exposé en dehors d'AWS KMS.

La valeur `ciphertextBlob` que vous spécifiez doit être la valeur du `CiphertextBlob` champ provenant d'une réponse GenerateDataKeyGenerateDataKeyWithoutPlaintext, ou Chiffrer, ou

le `PrivateKeyCiphertextBlob` champ d'une GenerateDataKeyPairWithoutPlaintextréponse GenerateDataKeyPairou. Vous pouvez également utiliser l'opération `ReEncrypt` pour rechiffrer les données chiffrées en dehors de AWS KMS par la clé publique dans une clé KMS asymétrique.

Le paramètre `SourceKeyId` n'est pas requis lors du rechiffrement avec des clés KMS de chiffrement symétriques. AWS KMS peut obtenir la clé KMS utilisée pour chiffrer les données à partir des métadonnées dans le blob de texte chiffré. Toutefois, la spécification de la clé KMS que vous utilisez est une bonne pratique. Cette pratique garantit que vous utilisez la clé KMS prévue et vous empêche de déchiffrer par inadvertance un texte chiffré à l'aide d'une clé KMS non fiable.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode reEncrypt dans la Référence d'API AWS SDK for Java.

```
// Re-encrypt a data key

ByteBuffer sourceCiphertextBlob = Place your ciphertext here;

// Replace the following example key ARNs with valid key identfiers
String sourceKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String destinationKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

ReEncryptRequest req = new ReEncryptRequest();
req.setCiphertextBlob(sourceCiphertextBlob);
req.setSourceKeyId(sourceKeyId);
req.setDestinationKeyId(destinationKeyId);
ByteBuffer destinationCipherTextBlob = kmsClient.reEncrypt(req).getCiphertextBlob();
```

C#

Pour obtenir des détails, consultez la méthode ReEncrypt dans AWS SDK for .NET.

```
// Re-encrypt a data key

MemoryStream sourceCiphertextBlob = new MemoryStream();
```

```
// Write ciphertext to memory stream

// Replace the following example key ARNs with valid key identfiers
String sourceKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String destinationKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

ReEncryptRequest reEncryptRequest = new ReEncryptRequest()
{
    CiphertextBlob = sourceCiphertextBlob,
    SourceKeyId = sourceKeyId,
    DestinationKeyId = destinationKeyId
};
MemoryStream destinationCipherTextBlob =
 kmsClient.ReEncrypt(reEncryptRequest).CiphertextBlob;
```

Python

Pour obtenir des détails, consultez la méthode `re_encrypt` dans AWS SDK for Python (Boto3).

```
# Re-encrypt a data key
ciphertext = 'Place your ciphertext here'

# Replace the following example key ARNs with valid key identfiers
source_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
destination_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

response = kms_client.re_encrypt(
    CiphertextBlob=ciphertext,
    SourceKeyId=source_key_id,
    DestinationKeyId=destination_key_id
)

destination_ciphertext_blob = response['CiphertextBlob']
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `re_encrypt` dans le kit AWS SDK for Ruby.

```
# Re-encrypt a data key

ciphertext = 'Place your ciphertext here'
ciphertext_packed = [ciphertext].pack("H*")

# Replace the following example key ARNs with valid key identfiers
source_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
destination_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

response = kmsClient.re_encrypt({
  ciphertext_blob: ciphertext_packed,
  source_key_id: source_key_id,
  destination_key_id: destination_key_id
})

destination_ciphertext_blob = response.ciphertext_blob.unpack('H*')
```

PHP

Pour obtenir des détails, consultez la méthode ReEncrypt dans AWS SDK for PHP.

```
// Re-encrypt a data key

$ciphertextBlob = 'Place your ciphertext here';

// Replace the following example key ARNs with valid key identfiers
$sourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$destinationKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';

$result = $KmsClient->reEncrypt([
    'CiphertextBlob' => $ciphertextBlob,
    'SourceKeyId' => $sourceKeyId,
    'DestinationKeyId' => $destinationKeyId,
]);
```

Node.js

Pour plus de détails, consultez la propriété ReEncrypt dans AWSle SDK JavaScript pour Node.js.

```
// Re-encrypt a data key
const CiphertextBlob = 'Place your cipher text blob here';
// Replace the following example key ARNs with valid key identfiers
const SourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const DestinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321';

kmsClient.reEncrypt({ CiphertextBlob, SourceKeyId, DestinationKeyId }, (err, data)
 => {
  ...
});
```

PowerShell

Pour rechiffrer un texte chiffré sous la même clé KMS ou une clé différente, utilisez l'applet de commande Invoke-KMS. ReEncrypt

Étant donné que cet exemple utilise le texte chiffré renvoyé par une applet de commande de chiffrement AWS KMS, il utilise un objet `MemoryStream` pour la valeur du paramètre `CiphertextBlob`. Cependant, le paramètre `CiphertextBlob` de `Invoke-KMSReEncrypt` prend un tableau d'octets (`byte[]`) ; il ne nécessite pas d'objet `MemoryStream`. À partir de AWSPowerShell la version 4.0, les paramètres de tous les AWSPowerShell modules qui acceptent des tableaux d'octets et `MemoryStream` des objets acceptent les tableaux d'octets, les `MemoryStream` objets, les chaînes, les tableaux de chaînes et `FileInfo` (System.IO). FileInfo) objets. Vous pouvez passer n'importe lequel de ces types à `Invoke-KMSReEncrypt`.

```
# Re-encrypt a data key

[System.IO.MemoryStream]$ciphertextBlob = Read-Host 'Place your cipher text blob
 here'

# Replace the following example key ARNs with valid key identfiers
$sourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$destinationKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

$response = Invoke-KMSReEncrypt -Ciphertext $ciphertextBlob -SourceKeyId
 $sourceKeyId -DestinationKeyId $destinationKeyId
$reEncryptedCiphertext = $response.CiphertextBlob
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Utilisation de politiques de clé

Les exemples de cette rubrique utilisent l'API AWS KMS pour afficher et modifier les politiques de clé des AWS KMS keys.

Pour plus de détails sur la façon d'utiliser les politiques de clé, les politiques IAM et les octrois pour gérer l'accès à vos clés KMS, veuillez consulter Authentification et contrôle d'accès pour AWS KMS. Pour obtenir de l'aide sur la rédaction et la mise en forme d'un document de politique JSON, consultez la Référence de politique JSON IAM dans le Guide de l'utilisateur IAM.

Rubriques

- Établissement de la liste des politiques de clé
- Obtention d'une politique de clé
- Définition d'une politique de clé

Établissement de la liste des politiques de clé

Pour obtenir les noms des politiques clés pour unAWS KMS key, utilisez l'ListKeyPoliciesopération. Le seul nom de politique de clé qu'elle renvoie est default.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour plus de détails sur l'implémentation de Java, consultez la listKeyPolicies méthode dans la référence de l'AWS SDK for JavaAPI.

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
```

```
ListKeyPoliciesRequest req = new ListKeyPoliciesRequest().withKeyId(keyId);
ListKeyPoliciesResult result = kmsClient.listKeyPolicies(req);
```

C#

Pour obtenir des détails, consultez la méthode ListKeyPolicies dans AWS SDK for .NET.

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListKeyPoliciesRequest listKeyPoliciesRequest = new ListKeyPoliciesRequest()
{
    KeyId = keyId
};
ListKeyPoliciesResponse listKeyPoliciesResponse =
 kmsClient.ListKeyPolicies(listKeyPoliciesRequest);
```

Python

Pour obtenir des détails, consultez la méthode `list_key_policies` dans AWS SDK for Python (Boto3).

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_key_policies(
    KeyId=key_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `list_key_policies` dans le kit AWS SDK for Ruby.

```
# List key policies
```

```
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_key_policies({
  key_id: key_id
})
```

PHP

Pour obtenir des détails, consultez la méthode ListKeyPolicies dans AWS SDK for PHP.

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->listKeyPolicies([
    'KeyId' => $keyId
]);
```

Node.js

Pour plus de détails, consultez la listKeyPolicies propriété dans le AWSSDK pour JavaScript dans Node.js.

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

kmsClient.listKeyPolicies({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour répertorier le nom de la politique de clé par défaut, utilisez l'applet de commande Get-KMS KeyPolicyList.

```
# List key policies
```

```
# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$response = Get-KMSKeyPolicyList -KeyId $keyId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Obtention d'une politique de clé

Pour obtenir la politique clé d'unAWS KMS key, utilisez l'GetKeyPolicyopération.

GetKeyPolicy nécessite un nom de politique. Le seul nom de politique valide est default.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour plus de détails, consultez la getKeyPolicy méthode dans la référence de AWS SDK for Java l'API.

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";

GetKeyPolicyRequest req = new
 GetKeyPolicyRequest().withKeyId(keyId).withPolicyName(policyName);
GetKeyPolicyResult result = kmsClient.getKeyPolicy(req);
```

C#

Pour obtenir des détails, consultez la méthode GetKeyPolicy dans AWS SDK for .NET.

```
// Get the policy for a KMS key
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";

GetKeyPolicyRequest getKeyPolicyRequest = new GetKeyPolicyRequest()
{
    KeyId = keyId,
    PolicyName = policyName
};
GetKeyPolicyResponse getKeyPolicyResponse =
 kmsClient.GetKeyPolicy(getKeyPolicyRequest);
```

Python

Pour obtenir des détails, consultez la méthode `get_key_policy` dans AWS SDK for Python (Boto3).

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'

response = kms_client.get_key_policy(
    KeyId=key_id,
    PolicyName=policy_name
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `get_key_policy` dans le kit AWS SDK for Ruby.

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
```

```
response = kmsClient.get_key_policy({
  key_id: key_id,
  policy_name: policy_name
})
```

PHP

Pour obtenir des détails, consultez la méthode GetKeyPolicy dans AWS SDK for PHP.

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$policyName = "default";

$result = $KmsClient->getKeyPolicy([
    'KeyId' => $keyId,
    'PolicyName' => $policyName
]);
```

Node.js

Pour plus de détails, consultez la getKeyPolicy propriété dans le AWSSDK pour JavaScript dans Node.js.

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const PolicyName = 'default';
kmsClient.getKeyPolicy({ KeyId, PolicyName }, (err, data) => {
  ...
});
```

PowerShell

Pour obtenir la politique de clé d'une clé KMS, utilisez l'applet de commande Get-KMS KeyPolicy. Cette applet de commande renvoie la politique clé sous forme de chaîne (System.String) que vous pouvez utiliser dans une commande KeyPolicyWrite-KMS (). `PutKeyPolicy` Pour convertir

les politiques de la chaîne JSON en `PSCustomObject` objets, utilisez l'applet de commande ConvertFrom-JSON.

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$policyName = 'default'

$response = Get-KMSKeyPolicy -KeyId $keyId -PolicyName $policyName
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Définition d'une politique de clé

Pour créer ou remplacer la politique de clé pour une clé KMS, utilisez l'PutKeyPolicyopération.

`PutKeyPolicy` nécessite un nom de politique. Le seul nom de politique valide est default.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour plus de détails, consultez la putKeyPolicy méthode dans la référence de AWS SDK for Java l'API.

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";
String policy = "{" +
                "  \"Version\": \"2012-10-17\"," +
                "  \"Statement\": [{" +
                "    \"Sid\": \"Allow access for ExampleRole\"," +
                "    \"Effect\": \"Allow\"," +
```

```
                // Replace the following example user ARN with a valid one
                "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/
ExampleKeyUserRole\"}," +
                "    \"Action\": [" +
                "      \"kms:Encrypt\"," +
                "      \"kms:GenerateDataKey*\"," +
                "      \"kms:Decrypt\"," +
                "      \"kms:DescribeKey\"," +
                "      \"kms:ReEncrypt*\"" +
                "    ]," +
                "    \"Resource\": \"*\"" +
                "  }]" +
                "}";

PutKeyPolicyRequest req = new
 PutKeyPolicyRequest().withKeyId(keyId).withPolicy(policy).withPolicyName(policyName);
kmsClient.putKeyPolicy(req);
```

C#

Pour obtenir des détails, consultez la méthode PutKeyPolicy dans AWS SDK for .NET.

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";
String policy = "{" +
                "  \"Version\": \"2012-10-17\"," +
                "  \"Statement\": [{" +
                "    \"Sid\": \"Allow access for ExampleUser\"," +
                "    \"Effect\": \"Allow\"," +
                // Replace the following example user ARN with a valid one
                "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/
ExampleKeyUserRole\"}," +
                "    \"Action\": [" +
                "      \"kms:Encrypt\"," +
                "      \"kms:GenerateDataKey*\"," +
                "      \"kms:Decrypt\"," +
                "      \"kms:DescribeKey\"," +
                "      \"kms:ReEncrypt*\"" +
                "    ]," +
                "    \"Resource\": \"*\"" +
```

```
                "  }]" +
                "}";

PutKeyPolicyRequest putKeyPolicyRequest = new PutKeyPolicyRequest()
{
    KeyId = keyId,
    Policy = policy,
    PolicyName = policyName
};
kmsClient.PutKeyPolicy(putKeyPolicyRequest);
```

Python

Pour obtenir des détails, consultez la méthode `put_key_policy` dans AWS SDK for Python (Boto3).

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
policy = """
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "Allow access for ExampleUser",
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
        "Action": [
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:ReEncrypt*"
        ],
        "Resource": "*"
    }]
}"""

response = kms_client.put_key_policy(
    KeyId=key_id,
    Policy=policy,
```

```
    PolicyName=policy_name
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance put_key_policy dans le kit AWS SDK for Ruby.

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
policy = "{" +
  "  \"Version\": \"2012-10-17\"," +
  "  \"Statement\": [{" +
  "    \"Sid\": \"Allow access for ExampleUser\"," +
  "    \"Effect\": \"Allow\"," +
  # Replace the following example user ARN with a valid one
  "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/ExampleKeyUserRole
\"}," +
  "    \"Action\": [" +
  "      \"kms:Encrypt\"," +
  "      \"kms:GenerateDataKey*\"," +
  "      \"kms:Decrypt\"," +
  "      \"kms:DescribeKey\"," +
  "      \"kms:ReEncrypt*\"" +
  "    ]," +
  "    \"Resource\": \"*\"" +
  "  }]" +
  "}"

response = kmsClient.put_key_policy({
  key_id: key_id,
  policy: policy,
  policy_name: policy_name
})
```

PHP

Pour obtenir des détails, consultez la méthode PutKeyPolicy dans AWS SDK for PHP.

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$policyName = "default";

$result = $KmsClient->putKeyPolicy([
    'KeyId' => $keyId,
    'PolicyName' => $policyName,
    'Policy' => '{
        "Version": "2012-10-17",
        "Id": "custom-policy-2016-12-07",
        "Statement": [
            { "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal":
               { "AWS": "arn:aws:iam::111122223333:user/root" },
            "Action": [ "kms:*" ],
            "Resource": "*" },
            { "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal":
               { "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole" },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*" }
        ]
    } '
]);
```

Node.js

Pour plus de détails, consultez la putKeyPolicy propriété dans le AWSSDK pour JavaScript dans Node.js.

```
// Set a key policy for a KMS key
//
```

```
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const PolicyName = 'default';
const Policy = `{
    "Version": "2012-10-17",
    "Id": "custom-policy-2016-12-07",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        }
    ]
}`; // The key policy document

kmsClient.putKeyPolicy({ KeyId, Policy, PolicyName }, (err, data) => {
  ...
});
```

PowerShell

Pour définir une politique de clé pour une clé KMS, utilisez l'applet de commande Write-KMS KeyPolicy. Cette applet de commande ne renvoie aucune sortie. Pour vérifier l'efficacité de la commande, utilisez l'applet de commande Get-KMS KeyPolicy.

Le paramètre `Policy` prend une valeur de chaîne. Placez la chaîne entre guillemets simples pour en faire une chaîne littérale. Vous n'avez pas besoin d'utiliser des caractères de continuation ou des caractères d'échappement dans la chaîne littérale.

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$policyName = 'default'
$policy = '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        }]
```

```
    }'

Write-KMSKeyPolicy -KeyId $keyId -PolicyName $policyName -Policy $policy
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Utilisation d'octrois

Les exemples de cette rubrique utilisent l'API AWS KMS pour créer, afficher, retirer et révoquer des octrois sur les AWS KMS keys. Pour de plus amples informations sur l'utilisation des octrois dans AWS KMS, veuillez consulter Octrois dans AWS KMS.

Rubriques

- Création d'un octroi
- Affichage d'un octroi
- Abandon d'un octroi
- Révocation d'un octroi

Création d'un octroi

Pour créer une subvention pour unAWS KMS key, utilisez l'CreateGrantopération. La réponse inclut uniquement l'ID d'octroi et le jeton d'octroi. Pour obtenir des informations détaillées sur la subvention, utilisez l'ListGrantsopération, comme indiqué dansAffichage d'un octroi.

Ces exemples créent une autorisation qui permet aux utilisateurs qui peuvent assumer le `ExampleKeyUser` rôle d'appeler l'GenerateDataKeyopération sur la clé KMS identifiée par le `KeyId` paramètre.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode createGrant dans la Référence d'API AWS SDK for Java.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
String operation = GrantOperation.GenerateDataKey.toString();

CreateGrantRequest request = new CreateGrantRequest()
    .withKeyId(keyId)
    .withGranteePrincipal(granteePrincipal)
    .withOperations(operation);

CreateGrantResult result = kmsClient.createGrant(request);
```

C#

Pour obtenir des détails, consultez la méthode CreateGrant dans AWS SDK for .NET.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
String operation = GrantOperation.GenerateDataKey;

CreateGrantRequest createGrantRequest = new CreateGrantRequest()
{
    KeyId = keyId,
    GranteePrincipal = granteePrincipal,
    Operations = new List<string>() { operation }
};

CreateGrantResponse createGrantResult = kmsClient.CreateGrant(createGrantRequest);
```

Python

Pour obtenir des détails, consultez la méthode `create_grant` dans AWS SDK for Python (Boto3).

```
# Create a grant
```

```
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee_principal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
operation = ['GenerateDataKey']

response = kms_client.create_grant(
    KeyId=key_id,
    GranteePrincipal=grantee_principal,
    Operations=operation
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `create_grant` dans le kit AWS SDK for Ruby.

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee_principal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
operation = ['GenerateDataKey']

response = kmsClient.create_grant({
  key_id: key_id,
  grantee_principal: grantee_principal,
  operations: operation
})
```

PHP

Pour obtenir des détails, consultez la méthode CreateGrant dans AWS SDK for PHP.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
$operation = ['GenerateDataKey']
```

```
$result = $KmsClient->createGrant([
    'GranteePrincipal' => $granteePrincipal,
    'KeyId' => $keyId,
    'Operations' => $operation
]);
```

Node.js

Pour plus de détails, consultez la propriété CreateGrant dans le AWSSDK pour JavaScript Node.js.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const GranteePrincipal = 'arn:aws:iam::111122223333:role/ExampleKeyUser';
const Operations: ["GenerateDataKey"];
kmsClient.createGrant({ KeyId, GranteePrincipal, Operations }, (err, data) => {
  ...
});
```

PowerShell

Pour créer un octroi, utilisez l'applet de commande New-KMSGrant.

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$granteePrincipal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
$operation = 'GenerateDataKey'

$response = New-KMSGrant -GranteePrincipal $granteePrincipal -KeyId $keyId -
Operation $operation
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Affichage d'un octroi

Pour obtenir des informations détaillées sur les autorisations associées à une clé KMS, utilisez l'ListGrantsopération.

> **Note**
>
> Le champ `GranteePrincipal` de la réponse `ListGrants` contient habituellement le bénéficiaire principal. Toutefois, lorsque le principal bénéficiaire de l'attribution est un service AWS, le champ `GranteePrincipal` contient le mandataire de service, qui peut représenter plusieurs bénéficiaires principaux.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Ces exemples utilisent le paramètre facultatif `Limits`, qui détermine le nombre d'octrois renvoyés par l'opération.

Java

Pour plus de détails sur l'implémentation Java, veuillez consulter la méthode listGrants dans la Référence d'API AWS SDK for Java.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
Integer limit = 10;

ListGrantsRequest req = new ListGrantsRequest().withKeyId(keyId).withLimit(limit);
ListGrantsResult result = kmsClient.listGrants(req);
```

C#

Pour obtenir des détails, consultez la méthode ListGrants dans AWS SDK for .NET.

```
// Listing grants on a KMS key
//
```

```
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
int limit = 10;

ListGrantsRequest listGrantsRequest = new ListGrantsRequest()
{
    KeyId = keyId,
    Limit = limit
};
ListGrantsResponse listGrantsResponse = kmsClient.ListGrants(listGrantsRequest);
```

Python

Pour obtenir des détails, consultez la méthode `list_grants` dans AWS SDK for Python (Boto3).

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_grants(
    KeyId=key_id,
    Limit=10
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `list_grants` dans le kit AWS SDK for Ruby.

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_grants({
  key_id: key_id,
  limit: 10
})
```

PHP

Pour obtenir des détails, consultez la méthode ListGrants dans AWS SDK for PHP.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$limit = 10;

$result = $KmsClient->listGrants([
    'KeyId' => $keyId,
    'Limit' => $limit,
]);
```

Node.js

Pour plus de détails, consultez la propriété ListGrants dans le AWSSDK pour JavaScript Node.js.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Limit = 10;
kmsClient.listGrants({ KeyId, Limit }, (err, data) => {
  ...
});
```

PowerShell

Pour afficher les détails de toutes les AWS KMS autorisations relatives à une clé KMS, utilisez l'applet de commande Get-KMS GrantList.

Pour limiter le nombre d'objets en sortie, cet exemple utilise l'applet de commande Select-Object au lieu du paramètre `Limit`, obsolète dans cette applet de commande. Pour obtenir de l'aide sur la pagination de sortie dans AWS Tools for PowerShell, veuillez consulter le billet de blog Output Pagination with AWS Tools for PowerShell.

```
# Listing grants on a KMS key
```

```
# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$limit = 10

$response = Get-KMSGrantList -KeyId $keyId | Select-Object -First $limit
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Vous devez spécifier la clé KMS dans chaque opération `ListGrants`. Toutefois, vous pouvez filtrer davantage la liste d'octrois en spécifiant l'ID d'octroi ou le principal bénéficiaire. Les exemples suivants obtiennent uniquement les octrois pour une clé KMS dans laquelle le rôle `test-engineer` est le principal bénéficiaire.

Java

Pour plus de détails sur l'implémentation Java, veuillez consulter la méthode listGrants dans la Référence d'API AWS SDK for Java.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String grantee = "arn:aws:iam::111122223333:role/test-engineer";

ListGrantsRequest req = new
 ListGrantsRequest().withKeyId(keyId).withGranteePrincipal(grantee);
ListGrantsResult result = kmsClient.listGrants(req);
```

C#

Pour obtenir des détails, consultez la méthode ListGrants dans AWS SDK for .NET.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
```

```
String grantee = "arn:aws:iam::111122223333:role/test-engineer";

ListGrantsRequest listGrantsRequest = new ListGrantsRequest()
{
    KeyId = keyId,
    GranteePrincipal = grantee
};
ListGrantsResponse listGrantsResponse = kmsClient.ListGrants(listGrantsRequest);
```

Python

Pour obtenir des détails, consultez la méthode `list_grants` dans AWS SDK for Python (Boto3).

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee = 'arn:aws:iam::111122223333:role/test-engineer'

response = kms_client.list_grants(
    KeyId=key_id,
    GranteePrincipal=grantee
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `list_grants` dans le kit AWS SDK for Ruby.

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee = 'arn:aws:iam::111122223333:role/test-engineer'

response = kmsClient.list_grants({
  key_id: keyId,
  grantee_principal: grantee
})
```

PHP

Pour obtenir des détails, consultez la méthode ListGrants dans AWS SDK for PHP.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$grantee = 'arn:aws:iam::111122223333:role/test-engineer';

$result = $KmsClient->listGrants([
    'KeyId' => $keyId,
    'GranteePrincipal' => $grantee,
]);
```

Node.js

Pour plus de détails, consultez la propriété ListGrants dans le AWSSDK pour JavaScript Node.js.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Grantee = 'arn:aws:iam::111122223333:role/test-engineer';

kmsClient.listGrants({ KeyId, Grantee }, (err, data) => {
  ...
});
```

PowerShell

Pour afficher les détails de toutes les AWS KMS autorisations relatives à une clé KMS, utilisez l'applet de commande Get-KMS GrantList.

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$grantee = 'arn:aws:iam::111122223333:role/test-engineer'
$response = Get-KMSGrantList -KeyId $keyId -GranteePrincipal $grantee
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Abandon d'un octroi

Pour annuler une autorisation pour une clé KMS, utilisez l'RetireGrantopération. Vous devez abandonner un octroi pour nettoyer une fois que vous avez fini de l'utiliser.

Pour retirer un octroi, fournissez le jeton d'octroi ou à la fois l'ID d'octroi et l'ID de clé KMS. Pour cette opération, l'ID de clé KMS doit être l'Amazon Resource Name (ARN) de la clé KMS. Le jeton de subvention est renvoyé par l'CreateGrantopération. L'ID de subvention est renvoyé par les ListGrantsopérations CreateGrant et.

RetireGrant ne renvoie pas de réponse. Pour vérifier son efficacité, utilisez l'ListGrantsopération.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode retireGrant dans la Référence d'API AWS SDK for Java.

```
// Retire a grant
//
String grantToken = Place your grant token here;

RetireGrantRequest req = new RetireGrantRequest().withGrantToken(grantToken);
kmsClient.retireGrant(req);
```

C#

Pour obtenir des détails, consultez la méthode RetireGrant dans AWS SDK for .NET.

```
// Retire a grant
//
String grantToken = "Place your grant token here";

RetireGrantRequest retireGrantRequest = new RetireGrantRequest()
{
```

```
    GrantToken = grantToken
};
kmsClient.RetireGrant(retireGrantRequest);
```

Python

Pour obtenir des détails, consultez la méthode `retire_grant` dans AWS SDK for Python (Boto3).

```
# Retire a grant

grant_token = Place your grant token here

response = kms_client.retire_grant(
    GrantToken=grant_token
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `retire_grant` dans le kit AWS SDK for Ruby.

```
# Retire a grant

grant_token = Place your grant token here

response = kmsClient.retire_grant({
  grant_token: grant_token
})
```

PHP

Pour obtenir des détails, consultez la méthode RetireGrant dans AWS SDK for PHP.

```
// Retire a grant
//
$grantToken = 'Place your grant token here';

$result = $KmsClient->retireGrant([
    'GrantToken' => $grantToken,
]);
```

Node.js

Pour plus de détails, consultez la propriété RetireGrant dans AWSle SDK JavaScript pour Node.js.

```
// Retire a grant
//
const GrantToken = 'Place your grant token here';
kmsClient.retireGrant({ GrantToken }, (err, data) => {
  ...
});
```

PowerShell

Pour retirer un octroi, utilisez l'applet de commande Disable-KMSGrant. Pour obtenir un jeton d'octroi, utilisez l'applet de commande New-KMSGrant. Le paramètre `GrantToken` prend une chaîne, vous n'avez donc pas besoin de convertir la sortie renvoyée par l'applet de commande Read-Host.

```
# Retire a grant

$grantToken = Read-Host -Message Place your grant token here
Disable-KMSGrant -GrantToken $grantToken
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Révocation d'un octroi

Pour révoquer une autorisation accordée à une clé KMS, utilisez l'RevokeGrantopération. Vous pouvez révoquer un octroi pour refuser explicitement les opérations qui en dépendent.

Dans les langues qui nécessitent un objet client, ces exemples utilisent l'objet client AWS KMS que vous avez créé dans Création d'un client.

Java

Pour obtenir des détails, veuillez consulter la méthode revokeGrant dans la Référence d'API AWS SDK for Java.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

// Replace the following example grant ID with a valid one
String grantId = "grant1";

RevokeGrantRequest req = new
 RevokeGrantRequest().withKeyId(keyId).withGrantId(grantId);
kmsClient.revokeGrant(req);
```

C#

Pour obtenir des détails, consultez la méthode RevokeGrant dans AWS SDK for .NET.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

// Replace the following example grant ID with a valid one
String grantId = "grant1";

RevokeGrantRequest revokeGrantRequest = new RevokeGrantRequest()
{
    KeyId = keyId,
    GrantId = grantId
};
kmsClient.RevokeGrant(revokeGrantRequest);
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Python

Pour obtenir des détails, consultez la méthode `revoke_grant` dans AWS SDK for Python (Boto3).

```
# Revoke a grant on a KMS key
```

```
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
grant_id = 'grant1'

response = kms_client.revoke_grant(
    KeyId=key_id,
    GrantId=grant_id
)
```

Ruby

Pour obtenir des détails, consultez la méthode d'instance `revoke_grant` dans le kit AWS SDK for Ruby.

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
grant_id = 'grant1'

response = kmsClient.revoke_grant({
  key_id: key_id,
  grant_id: grant_id
})
```

PHP

Pour obtenir des détails, consultez la méthode RevokeGrant dans AWS SDK for PHP.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

// Replace the following example grant ID with a valid one
```

```
$grantId = "grant1";

$result = $KmsClient->revokeGrant([
    'KeyId' => $keyId,
    'GrantId' => $grantId,
]);
```

Node.js

Pour plus de détails, consultez la propriété RevokeGrant dans AWSle SDK JavaScript pour Node.js.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

// Replace the following example grant ID with a valid one
const GrantId = 'grant1';
kmsClient.revokeGrant({ GrantId, KeyId }, (err, data) => {
  ...
});
```

PowerShell

Pour révoquer un octroi, utilisez l'applet de commande Revoke-KMSGrant.

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
$grantId = 'grant1'

Revoke-KMSGrant -KeyId $keyId -GrantId $grantId
```

Pour utiliser les AWS KMS PowerShell applets de commande, installez le fichier AWS.tools.KeyManagementServicemodule. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS Tools for Windows PowerShell.

Tester vos appels d'API AWS KMS

Pour utiliser AWS KMS, vous devez posséder des informations d'identification que AWS peut utiliser pour authentifier vos demandes d'API. Les informations d'identification doivent inclure des autorisations pour accéder aux clés KMS et aux alias. Les autorisations sont déterminées par les stratégies de clé, les politiques IAM, les octrois et les contrôles d'accès intercomptes. Outre le contrôle de l'accès aux clés KMS, vous pouvez contrôler l'accès à votre CloudHSM et à vos magasins de clés personnalisés.

Vous pouvez spécifier le paramètre d'API `DryRun` pour vérifier que vous disposez des autorisations nécessaires pour utiliser les clés AWS KMS. Vous pouvez également utiliser `DryRun` pour vérifier que les paramètres de demande dans un appel d'API AWS KMS sont correctement spécifiés.

Rubriques

- Quel est le DryRun paramètre ?
- Spécification DryRun à l'aide de l'API

Quel est le DryRun paramètre ?

`DryRun` est un paramètre d'API facultatif que vous spécifiez pour vérifier que les appels d'API AWS KMS aboutiront. Utilisez `DryRun` pour tester votre appel d'API, avant de passer réellement l'appel à AWS KMS. Vous pouvez modifier les valeurs suivantes :

- Que vous disposez des autorisations nécessaires pour utiliser les clés AWS KMS.
- Que vous avez correctement spécifié les paramètres lors de l'appel.

AWS KMS prend en charge l'utilisation du paramètre `DryRun` dans certaines actions d'API :

- CreateGrant
- Decrypt (Déchiffrer)
- Encrypt (Chiffrer)
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext

- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign (Signer)
- Verify (Vérifier)
- VerifyMac

L'utilisation du paramètre `DryRun` entraînera des frais et sera facturée comme une demande d'API standard. Pour plus d'informations sur la tarification AWS KMS, consultez Tarification AWS Key Management Service.

Toutes les demandes d'API utilisant le paramètre `DryRun` s'appliquent au quota de demandes de l'API et peuvent entraîner une exception de limitation si vous dépassez un quota de demandes d'API. Par exemple, le fait d'appeler Decrypt avec `DryRun` ou sans `DryRun` compte pour le même quota d'opérations cryptographiques. Pour en savoir plus, veuillez consulter Limitation des demandes AWS KMS.

Chaque appel à une opération d'API AWS KMS est capturé comme événement dans un journal AWS CloudTrail. Le résultat de toutes les opérations qui spécifient le `DryRun` paramètre apparaît dans votre CloudTrail journal. Pour plus d'informations, consultez Journalisation des appels d'API AWS KMS avec AWS CloudTrail.

Spécification DryRun à l'aide de l'API

Pour utiliser `DryRun`, spécifiez le paramètre `–dry-run` dans les commandes AWS CLI et les appels d'API AWS KMS qui prennent en charge le paramètre. Lorsque vous le ferez, AWS KMS vérifiera si votre appel aboutit. Les appels AWS KMS qui utilisent `DryRun` échoueront toujours et renverront un message contenant des informations sur le motif d'échec de l'appel. Le message peut inclure les exceptions suivantes :

- `DryRunOperationException` - La demande aboutirait si `DryRun` n'était pas spécifié.
- `ValidationException` - La demande n'a pas réussi à spécifier un paramètre d'API incorrect.
- `AccessDeniedException` - Vous ne disposez pas des autorisations pour exécuter l'action d'API spécifiée sur la ressource KMS.

Par exemple, la commande suivante utilise l'CreateGrantopération et crée une autorisation qui permet aux utilisateurs autorisés à assumer le `keyUserRole` rôle d'appeler l'opération de déchiffrement sur une clé KMS symétrique spécifiée. Le paramètre `DryRun` est spécifié.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

cohérence à terme AWS KMS

L'API AWS KMS suit un modèle de cohérence à terme en raison de la nature distribuée du système. Par conséquent, les modifications apportées aux ressources AWS KMS peuvent ne pas être immédiatement visibles pour les commandes suivantes que vous exécuterez.

Lorsque vous effectuez des appels d'API AWS KMS, il se peut qu'il y ait un bref délai avant que le changement ne soit disponible via AWS KMS. La propagation de la modification dans l'ensemble du système prend généralement moins de quelques secondes, mais dans certains cas, cela peut prendre plusieurs minutes. Vous risquez de recevoir des erreurs inattendues, telles qu'un `NotFoundException` ou un `InvalidStateException`, pendant cette période. Par exemple, AWS KMS peut renvoyer un `NotFoundException` si vous appelez `GetParametersForImport` immédiatement après avoir appelé `CreateKey`.

Nous vous recommandons de configurer une stratégie de relance pour vos clients AWS KMS afin de relancer automatiquement les opérations après une brève période d'attente. Pour en savoir plus, consultez Comportement de nouvelle tentative dans le Guide de références des kits SDK et des outils AWS.

Pour les appels d'API liés à un octroi, vous pouvez utiliser un jeton d'octroi pour éviter tout retard potentiel et utiliser immédiatement les autorisations d'un octroi. Pour plus d'informations, consultez la rubrique relative à la cohérence à terme (pour les octrois).

Références

Les références suivantes fournissent des informations utiles sur l'utilisation et la gestion des clés KMS.

- Référence de type de clé. Répertorie le type de clé KMS qui prend en charge chaque opération d'API AWS KMS

  Pour trouver : Puis-je activer et désactiver une clé KMS de signature RSA ?

- Tableau d'états de clé Indique comment l'état de clé d'une clé KMS affecte son utilisation dans les opérations d'API AWS KMS

  Pour trouver : Puis-je modifier l'alias d'une clé KMS en attente de suppression ?

- Référence des autorisations d'API AWS KMS Fournit des informations sur les autorisations requises pour chaque opération d'API AWS KMS

  À rechercher : Puis-je utiliser GetKeyPolicyune clé d'un autre AWS compte ? Puis-je octroyer l'autorisation `kms:Decrypt` dans une politique IAM ?

  - ViaService référence. Répertorie les services AWS prenant en charge la clé de condition `kms:ViaService`.

    À rechercher : puis-je utiliser la clé de `kms:ViaService` condition pour autoriser une autorisation uniquement lorsqu'elle provient d'Amazon ElastiCache ? Qu'en est-il d'Amazon Neptune ?

- Tarification AWS KMS. Répertorie et explique le prix des clés KMS.

  Pour trouver : Combien coûte l'utilisation de mes clés asymétriques ?

- Quotas de demande AWS KMS Répertorie les quotas par seconde des demandes d'API AWS KMS dans chaque compte et chaque région.

  Pour trouver : Combien de demandes Decrypt puis-je exécuter chaque seconde ? Combien de demandes Decrypt puis-je exécuter sur des clés KMS dans mon magasin de clés personnalisé ?

- Quotas de ressources AWS KMS Répertorie les quotas sur les ressources AWS KMS.

  Pour trouver : De combien de clés KMS puis-je disposer dans chaque région de mon compte ? De combien d'alias puis-je disposer sur chaque clé KMS ?

- Services AWS intégrés à AWS KMS Répertorie les services AWS qui utilisent des clés KMS pour protéger les ressources qu'ils créent, stockent et gèrent.

  Pour trouver : Amazon Connect utilise-t-il des clés KMS pour protéger mes ressources Connect ?

Historique du document

Cette rubrique décrit les mises à jour importantes apportées au Guide du développeur AWS Key Management Service.

Rubriques

- Mises à jour récentes
- Mises à jour antérieures

Mises à jour récentes

Le tableau suivant décrit les modifications importantes apportées à cette documentation depuis janvier 2018. En plus des principales modifications répertoriées ici, nous mettons fréquemment à jour la documentation pour améliorer les descriptions et les exemples, et pour répondre aux commentaires que vous nous envoyez. Pour recevoir une notification concernant des modifications importantes, abonnez-vous au flux RSS.

Il peut être nécessaire de faire défiler horizontalement ou verticalement pour afficher toutes les données de ce tableau.

Modification	Description	Date
Mises à jour de la politique gérée	De nouvelles autorisations ont été ajoutées `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` qui AWS KMS permettent de surveiller les modifications apportées au VPC qui contient votre AWS CloudHSM cluster afin de fournir des messages d'erreur clairs en cas de défaillance. AWS KMS	10 novembre 2023

Mise à jour de la fonctionnalité	Ajout de la prise en charge du paramètre d'API `DryRun`.	5 juillet 2023
Mise à jour de la fonctionnalité	Ajout de la prise en charge de l'importation de l'élément clé pour tous les types de clés AWS KMS, à l'exception des magasins de clés personnalisés.	5 juin 2023
Mise à jour de la fonctionnalité	Mises à jour des API AWS KMS pour les enclaves Nitro	10 mars 2023
Mise à jour de la fonctionnalité	L'algorithme d'encapsulage `RSAES_PKCS1_V1_5` est obsolète. AWS KMS mettra fin à tout support pour `RSAES_PKCS1_V1_5` d'ici le 1er octobre 2023 conformément aux directives de gestion des clés cryptographiques du National Institute of Standards and Technology (NIST). Nous vous recommandons de commencer à utiliser un algorithme d'encapsulage différent immédiatement.	28 février 2023
Mise à jour de la fonctionnalité	Ajout de la prise en charge des magasins de clés externes, une fonctionnalité qui vous permet de protéger vos ressources AWS en utilisant des clés cryptographiques en dehors d'AWS.	29 novembre 2022

Changement de quota	Augmentation de la valeur du quota de ressources AWS KMS keys à 100 000 clés pour chaque compte et région.	8 juillet 2022
Mise à jour des fonctionnalités	Ajout de la prise en charge des clés KMS HMAC dans plus de Régions AWS	8 juillet 2022
Nouvelle rubrique	Ajout de Résilience dans la rubrique AWS Key Management Service au chapitre Sécurité du AWS KMS Guide du développeur.	14 juin 2022
Nouvelle fonction	Ajout de la prise en charge de clés AWS KMS et opérations d'API qui génèrent et vérifient les codes HMAC.	19 avril 2022
Modification de la documentation	Remplacez le terme clé principale client (CMK) par AWS KMS key et clé KMS.	30 août 2021
Nouvelle fonction	Ajout de la prise en charge des clés multi-région, un ensemble de clés KMS interopérables dans différentes régions qui ont le même ID de clé et les mêmes éléments de clé. Vous pouvez utiliser des clés multi-région pour chiffrer des données dans une région et déchiffrer des données dans une autre région.	8 juin 2021

Nouvelle fonction	Ajout de la prise en charge du contrôle d'accès basé sur l'attribut (ABAC). Vous pouvez utiliser des balises et des alias afin de contrôler l'accès à vos AWS KMS keys.	17 décembre 2020
Nouvelle fonction	Ajout de la prise en charge pour les politiques de point de terminaison d'un VPC.	9 juillet 2020
Nouveau contenu	Explique les propriétés de sécurité de AWS KMS.	18 juin 2020
Nouvelle fonction	Ajout de la prise en charge AWS KMS keys asymétriq ues et des clés de données asymétriques.	25 novembre 2019
Fonction mise à jour	Vous pouvez afficher la politique de clé des Clés gérées par AWS dans la console AWS KMS. Cette fonction était auparavant limitée aux clés gérées par le client.	15 novembre 2019
Nouvelle fonction	Explique comment utiliser des algorithmes échange de clés post-quantiques hybrides dans TLS pour vos appels vers AWS KMS.	4 novembre 2019
Changement de quota	Augmentation des quotas de ressources pour certaines API qui gèrent les clés KMS.	18 septembre 2019

Changement de quota	Modification des quotas de ressources pour les clés KMS, les alias et les octrois par clé KMS.	27 mars 2019
Changement de quota	Modification du quota de demande par seconde partagé pour les opération s de chiffrement qui utilisent les AWS KMS keys dans un magasin de clés personnalisé.	7 mars 2019
Nouvelle fonction	Explique comment créer et gérer des AWS KMS magasins de clé personnal isés. Chaque magasin de clés est soutenu par un cluster AWS CloudHSM que vous possédez et contrôlez.	26 novembre 2018
Nouvelle console	Explique comment utiliser la nouvelle console AWS KMS, qui est indépendante de la console IAM. La console d'origine, et les instructi ons pour l'utiliser, resteront disponibles pendant une brève période pour vous donner le temps de vous familiariser avec la nouvelle console.	7 novembre 2018
Changement de quota	Modification du quota de demande partagé pour l'utilisa tion de AWS KMS keys.	21 août 2018

Nouveau contenu	Explique comment AWS Secrets Manager utilise les clés AWS KMS pour chiffrer la valeur du secret dans un secret.	13 juillet 2018
Nouveau contenu	Explique comment DynamoDB utilise les AWS KMS AWS KMS keys pour prendre en charge son option de chiffrement côté serveur.	23 mai 2018
Nouvelle fonction	Explique comment vous pouvez utiliser un point de terminaison privé dans votre VPC pour vous connecter directement à AWS KMS au lieu de vous connecter via Internet.	22 janvier 2018

Mises à jour antérieures

Le tableau suivant décrit les modifications importantes apportées au Guide du développeur AWS Key Management Service avant 2018.

Il peut être nécessaire de faire défiler horizontalement ou verticalement pour afficher toutes les données de ce tableau.

Modification	Description	Date
Nouveau contenu	Ajout de la documentation sur Clés de balisage.	15 février 2017
Nouveau contenu	Ajout de la documentation sur Surveillance des AWS KMS keys et Surveillance avec Amazon CloudWatch.	31 août 2016

Modification	Description	Date
Nouveau contenu	Ajout de la documentation sur Éléments de clé importés.	11 août 2016
Nouveau contenu	Ajout de la documentation suivante : Politiques IAM, Référence des autorisations et Clés de condition.	5 juillet 2016
Mettre à jour	Mise à jour de certaines parties de la documentation dans le chapitre Authentification et contrôle d'accès.	5 juillet 2016
Mettre à jour	Mise à jour de la page Quotas pour refléter les nouveaux quotas par défaut.	31 mai 2016
Mettre à jour	Mise à jour de la page Quotas pour refléter les nouveaux quotas par défaut et mise à jour de la documentation sur le jeton d'octroi pour améliorer la clarté et la précision.	11 avril 2016
Nouveau contenu	Ajout de la documentation sur Attribution à plusieurs principaux IAM de l'autorisation d'accès à une clé KMS et Utilisation de la condition d'adresse IP.	17 février 2016
Mettre à jour	Mise à jour des pages Politiques de clé dans AWS KMS et Modification d'une politique de clé pour améliorer la clarté et la précision.	17 février 2016

Modification	Description	Date
Mettre à jour	Mise à jour des pages de la rubrique Gestion de clés pour améliorer la clarté.	5 janvier 2016
Nouveau contenu	Ajout de la documentation sur Comment AWS CloudTrail utilise AWS KMS.	18 novembre 2015
Nouveau contenu	Ajout d'instructions pour Modification d'une politique de clé.	18 novembre 2015
Mettre à jour	Mise à jour de la documentation sur Comment Amazon Relational Database Service (Amazon RDS) utilise AWS KMS.	18 novembre 2015
Nouveau contenu	Ajout de la documentation sur Comment WorkSpaces utilise AWS KMS.	6 novembre 2015
Mettre à jour	Mise à jour de la page Politiques de clé dans AWS KMS pour améliorer la clarté.	22 octobre 2015
Nouveau contenu	Ajout de la documentation sur Suppression de AWS KMS keys, y compris de la documentation de prise en charge sur Création d'une alarme et Déterminer l'utilisation passée d'une clé KMS.	15 octobre 2015

Modification	Description	Date
Nouveau contenu	Ajout de la documentation sur Déterminer l'accès à des AWS KMS keys.	15 octobre 2015
Nouveau contenu	Ajout de la documentation sur Principaux états des clés AWS KMS.	15 octobre 2015
Nouveau contenu	Ajout de la documentation sur Comment Amazon Simple Email Service (Amazon SES) utilise AWS KMS.	1er octobre 2015
Mettre à jour	Mise à jour de la page Quotas pour expliquer les nouveaux quotas de demande.	31 août 2015
Nouveau contenu	Ajout d'informations sur les frais d'utilisation d'AWS KMS. Veuillez consulter Tarification AWS KMS.	14 août 2015
Nouveau contenu	Ajout de quotas de demande aux Quotas AWS KMS.	11 juin 2015
Nouveau contenu	Ajout d'un nouvel exemple de code Java illustrant l'utilisation de l'opération `UpdateAlias`. veuillez consulter Mise à jour d'un alias.	1er juin 2015
Mettre à jour	Déplacement de la table des régions AWS Key Management Service vers la Références générales AWS.	29 mai 2015

Modification	Description	Date
Nouveau contenu	Ajout de la documentation sur Comment Amazon EMR utilise AWS KMS.	28 janvier 2015
Nouveau contenu	Ajout de la documentation sur Comment Amazon WorkMail utilise AWS KMS.	28 janvier 2015
Nouveau contenu	Ajout de la documentation sur Comment Amazon Relational Database Service (Amazon RDS) utilise AWS KMS.	6 janvier 2015
Nouveau contenu	Ajout de la documentation sur Comment Amazon Elastic Transcoder utilise AWS KMS.	24 novembre 2014
Nouveau guide	Présentation du Guide du développeur AWS Key Management Service.	12 novembre 2014

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